



UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Rastreamento de Células em Vídeos 3D

Dissertação de Mestrado

Davy Oliveira Barros Sousa



São Cristóvão – Sergipe

2018

UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Davy Oliveira Barros Sousa

Rastreamento de Células em Vídeos 3D

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de mestre em Ciência da Computação.

Orientador(a): Prof. Dr. Hendrik Teixeira Macedo

São Cristóvão – Sergipe

2018

**FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL
UNIVERSIDADE FEDERAL DE SERGIPE**

S726r Sousa, Davy Oliveira Barros
Rastreamento de células em vídeos 3D / Davy Oliveira Barros
Sousa ; orientador Hendrik Teixeira Macedo . - São Cristóvão,
2018.
71 f. : il.

Dissertação (mestrado em Ciência da Computação) –
Universidade Federal de Sergipe, 2018.

1. Computação. 2. Processamento de imagens – técnicas
digitais. 3. Ciências médicas 4. Células. I. Macedo, Hendrik
Teixeira orient. II. Título.

CDU 004:611.018.1




UNIVERSIDADE FEDERAL DE SERGIPE
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA
COORDENAÇÃO DE PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

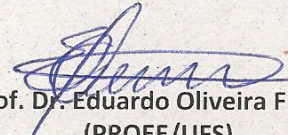
Ata da Sessão Solene de Defesa da Dissertação do
Curso de Mestrado em Ciência da Computação-UFS.
Candidato: DAVY OLIVEIRA BARROS SOUSA

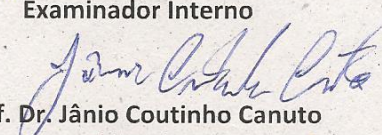
Em 28 dias do mês de agosto do ano de dois mil e dezoito, com início às 09h00min, realizou-se na Sala de Seminário do DCOMP da Universidade Federal de Sergipe, na Cidade Universitária Prof. José Aloísio de Campos, a Sessão Pública de Defesa de Dissertação de Mestrado do candidato **DAVY OLIVEIRA BARROS SOUSA**, que desenvolveu o trabalho intitulado: "**Rastreamento de Células em Vídeos 3D**", sob a orientação do Prof. Dr. **Hendrik Teixeira Macedo**. A Sessão foi presidida pelo Prof. Dr. **Hendrik Teixeira Macedo** (PROCC/UFS), que após a apresentação da dissertação passou a palavra aos outros membros da Banca Examinadora, Prof. Dr. **Eduardo Oliveira Freire** (PROEE/UFS), Prof. Dr. **Jugurta Rosa Montalvão Filho** (PROCC/UFS) e, em seguida, ao Prof. Dr. **Jânio Coutinho Canuto** (UFS). Após as discussões, a Banca Examinadora reuniu-se e considerou o mestrando (a) APROVADO "(aprovado/reprovado)" COM "(com/sem)" ressalvas. Atendidas as exigências da Instrução Normativa 01/2017/PROCC, do Regimento Interno do PROCC (Resolução 67/2014/CONEPE), e da Resolução nº 25/2014/CONEPE que regulamentam a Apresentação e Defesa de Dissertação, e nada mais havendo a tratar, a Banca Examinadora elaborou esta Ata que será assinada pelos seus membros e pelo mestrando.

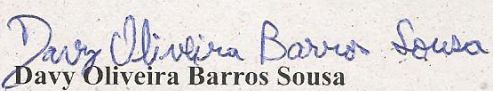
Cidade Universitária "Prof. José Aloísio de Campos", 28 de Agosto de 2018.


Prof. Dr. Hendrik Teixeira Macedo
(PROCC/UFS)
Presidente


Prof. Dr. Jugurta Rosa Montalvão Filho
(PROCC/UFS)
Examinador Interno


Prof. Dr. Eduardo Oliveira Freire
(PROEE/UFS)
Examinador Interno


Prof. Dr. Jânio Coutinho Canuto
(UFS)
Examinador Externo


Davy Oliveira Barros Sousa
Candidato

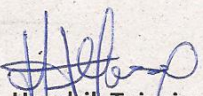


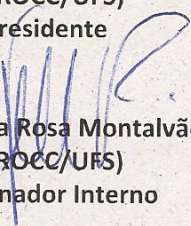


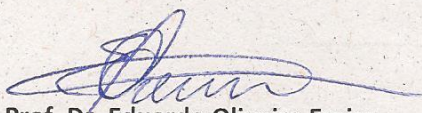
UNIVERSIDADE FEDERAL DE SERGIPE
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA
COORDENAÇÃO DE PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

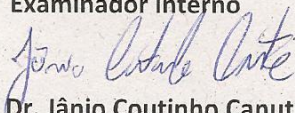
Observações (em caso de aprovação com ressalvas):

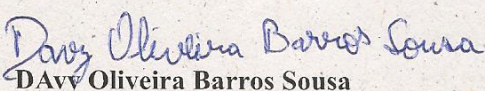
- 1-MODELAGEM MATEMÁTICA DA MOTILIDADE CELULAR E DAS POSSÍVEIS INTERAÇÕES COM O PROBLEMA DE PASTOREAMENTO
- 2-DESCRIÇÃO DA METODOLOGIA UTILIZADA PARA DETERMINAR AS OPERAÇÕES DE SEGMENTAÇÃO
- 3-LEVANTAR HIPÓTESES (E TESTÁ-LAS) DO PORQUÊ O MÉTODO FUNCIONA MELHOR EM 3D VS. 2D
- 4-REFLETIR NO DOCUMENTO FINAL TODAS CONSIDERAÇÕES DOCUMENTADAS PELA BANCA NO TEXTO.


Prof. Dr. Hendrik Teixeira Macedo
(PROCC/UFS)
Presidente


Prof. Dr. Jugurta Rosa Montalvão Filho
(PROCC/UFS)
Examinador Interno


Prof. Dr. Eduardo Oliveira Freire
(PROEE/UFS)
Examinador Interno


Prof. Dr. Jânio Coutinho Canuto
(UFS)
Examinador Externo


Davy Oliveira Barros Sousa
Candidato

Agradecimentos

Agradeço primeiramente a Deus por sempre me guiar, me mostrando o melhor caminho a ser tomado e por colocar pessoas maravilhosas em meu caminho para me ajudar em cada uma das minhas jornadas.

Agradeço aos meus pais José Bonifácio e Maria das Graças, por serem meus exemplos e sempre estarem ao meu lado, assim como aos meus irmãos Diogo e Viviany, que sempre me apoiam. Como minha mãe sempre diz: “Somos 5 corpos, mas uma só alma”. A conquista de um é a conquista de todos. Obrigado por serem o meu porto seguro.

Agradeço também a toda a minha família, tios, primos, avós, por estarem sempre torcendo pelo meu sucesso. Em especial, aos meus padrinhos, Gladstone e Juvenice, por acompanharem meu crescimento desde pequeno sempre vibrando, apoiando e torcendo por cada conquista minha.

Aos amigos, e a lista aqui é grande:

Aos meus amigos do 4Minute Brasil: Fabi, Gih, Rafa, Breno e João. Uma amizade que começou pelo simples fato de gostarmos do mesmo tipo de música, mas que com o passar do tempo virou uma amizade para a vida. Muito obrigado por serem essas pessoas maravilhosas, que me ouviam, me apoiavam e me ajudavam no que fosse preciso, muitas vezes não fazendo a mínima ideia do que eu estava falando.

Aos amigos que conquistei durante toda a minha vida acadêmica. Os “Bribos”, aqueles amigos da época do colegial e que levo para a vida. Obrigado por serem esses seres maravilhosos e que torcem pelo meu sucesso.

Aos amigos que conquistei durante a minha estadia na UFS, que estiveram acompanhando o meu desenvolvimento e me apoiando a cada passo. Em especial a: Larissa, Edinho, Augustto, Vanessa, Verônica, Dani, Léo Macêdo, Rose, Léo de Jesus, Ítalo, Alice, Diogo, Thiago, Elayne Torres. Muito obrigado por terem feito parte dessa minha trajetória.

Aos amigos do X-Mestrado: Marianne, Breno, Cícero, Erick, Denise e Thauane. Pessoas de astral ímpar com os quais tive o prazer de estudar. Grupo de amigos com o qual podíamos sempre contar para o que der e vier, dividindo os momentos de alegrias e de perrengues, sempre tentando ajudar uns aos outros no que podia. Como Mari sempre dizia: “Nem só de lattes vive o mestrado”. E fico muito feliz de que vocês tenham feito parte dessa minha história e quem sabe um dia do lattes.

À família Dcomp: Elaine Palmeira, Erickson, Elaine Silva, Aninha, Dina, Sara, George, Elder, Luan, e a muitos outros que fizeram parte dessa família mesmo que por apenas um certo tempo. A todos vocês que fazem esse departamento acontecer e tornaram os meus dias durante

minha estadia aí os melhores, meu muitíssimo obrigado pela amizade de vocês e por todo o apoio.

A todos os outros amigos que fui conquistando ao longo dessa jornada do mestrado como Lygia, Alef, Ademir, entre muitos outros. Assim como a todos os outros amigos que fui fazendo nesse meio tempo fora do meio acadêmico e que estiveram me apoiando e me mandando vibrações positivas, meu muito obrigado.

Agradeço também ao corpo docente do DCOMP e do PROCC pelos ensinamentos e conhecimentos compartilhados. Em especial, aos professores Ricardo Salgueiro, Gilton Mal, Kenia Kodel, e ao professor do PROEE Eduardo Freire pela ajuda na reta final do meu mestrado. Serei eternamente grato a cada um de vocês. Serei eternamente grato também ao professor Hendrik Macedo por ter aceitado ser o meu orientador aos 45 do segundo tempo, dando-me a possibilidade de mostrar o meu trabalho e assim me tornar mestre.

Agradeço também à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pela concessão da bolsa durante o período de realização deste mestrado.

Enfim, a todos vocês que contribuíram de alguma forma para o meu crescimento tanto profissional quanto pessoal, meu muito obrigado por todo o apoio.

Resumo

Realizar o rastreamento de células é importante para a cura e prevenção de doenças. Isso se deve ao fato da motilidade celular estar ligada a vários processos celulares. Porém, realizar a análise da trajetória de uma célula não é um trabalho simples, pois para armazenar as informações da trajetória é necessário uma grande quantidade de imagens, principalmente em imagens 3D. Dessa forma, faz-se necessária a criação de algoritmos que possam realizar o rastreamento celular de forma prática e automática. Neste trabalho foi dada continuidade ao trabalho de conclusão de curso (TCC) de Sousa (2015) de forma a melhorar o algoritmo de rastreamento já implementado e apresentado em Sousa (2015). Para suprir as deficiências do algoritmo implementado por Sousa (2015), foram feitas modificações na fase de segmentação, de rotulação e de rastreamento. Além disso, foram adicionados métodos para a detecção de divisão celular e a entrada e saída de novas células do vídeo. A validação do algoritmo foi feita por meio de um software de avaliação que possui rotinas tanto para a parte de segmentação quanto para a de rastreamento. Foram utilizados 11 *datasets* para a validação, sendo que o algoritmo proposto conseguiu obter resultados para 7 deles. Dentre os 7 *datasets*, 3 eram 2D e os outros 4, 3D. Embora os resultados não tenham sido satisfatórios para os *datasets* 2D, os 3D obtiveram resultados satisfatórios durante a fase de rastreamento, com médias de acurácia variando de 90,1% a 94,4%. A partir da validação, percebeu-se que, mesmo tendo obtido valores satisfatórios, inclusive alguns deles melhores que alguns encontrados no estado da arte, o algoritmo ainda precisa de melhorias.

Palavras-chave: Rastreamento de células, Imagens 3D, Processamento de imagens, Biomedicina.

Abstract

Performing cell tracking is important for curing and preventing diseases. This is due to the fact that cellular motility is linked to several cellular processes. However, performing the analysis of the trajectory of a cell is not a simple job, because to store the information of the trajectory it is necessary a great amount of images, mainly in 3D images. Thus, it is necessary to create algorithms that can perform cell tracking in a practical and automatic way. In this work was given a continuation of the work of conclusion of course (TCC) of [Sousa \(2015\)](#) in order to improve the tracking algorithm already implemented and presented in [Sousa \(2015\)](#). To overcome the shortcomings of the algorithm implemented by [Sousa \(2015\)](#), modifications were made to the segmentation, labeling and tracking phases. In addition, methods have been added for the detection of cell division and the entry and exit of new cells of the video. The validation of the algorithm was done through an evaluation software that has routines for both the segmentation and the tracking part. 11 datasets were used for the validation, and the proposed algorithm was able to obtain results for 7 of them. Among the 7 datasets, 3 were 2D and the other 4, 3D. Although the results were not satisfactory for the 2D datasets, the 3Ds obtained satisfactory results during the tracking phase, with averages of accuracy ranging from 90.1% to 94.4%. From the validation, it was realized that, even having obtained satisfactory values, including some of them better than some found in the state of the art, the algorithm still needs improvements.

Keywords: Cell tracking, 3D Images, Image processing, Biomedicine.

Lista de ilustrações

Figura 1 – Modelo idealizado da forma de um ceratócito se movendo de acordo com o modelo GRE	19
Figura 2 – Simulação do modelo GRE	20
Figura 3 – Execução das Rotinas de Avaliação da Segmentação e do Rastreamento . . .	27
Figura 4 – <i>Ground Truth</i> do <i>Dataset</i> Fluo-C3DL-MDA231	28
Figura 5 – Histograma da imagem “t000” do <i>dataset</i> Fluo-C3DL-MDA231	47
Figura 6 – Resultado da segmentação do <i>frame</i> 9 da imagem “t000” do <i>dataset</i> Fluo-C3DL-MDA231 antes do ajuste do <i>background</i>	48
Figura 7 – <i>Dataset</i> Fluo-C3DL-MDA231	49
Figura 8 – <i>Dataset</i> Fluo-C3DH-H157	50
Figura 9 – <i>Dataset</i> Fluo-N3DH-SIM	51
Figura 10 – <i>Dataset</i> Fluo-N3DH-CHO	52
Figura 11 – <i>Dataset</i> Fluo-C2DL-MSK	53
Figura 12 – <i>Dataset</i> Fluo-N2DH-GOWT1	53
Figura 13 – <i>Dataset</i> PhC-C2DH-U373	54
Figura 14 – Detecção de Divisão Celular	58
Figura 15 – Representação das distâncias do FoI	70
Figura 16 – Imagens Representativas dos <i>Datasets</i>	72

Lista de tabelas

Tabela 1 – Correspondência entre operações e pesos.	30
Tabela 2 – Comparação Entre os Métodos Utilizados Pelos Competidores do <i>Cell Tracking Challenge</i>	34
Tabela 3 – Resultados Obtidos pelo Algoritmo Proposto	60
Tabela 4 – Comparação Entre os Resultados do Algoritmo Proposto e dos Competidores do <i>Cell Tracking Challenge</i>	62
Tabela 5 – Características dos <i>Datasets</i>	71

Sumário

1	Introdução	13
1.1	Contextualização	13
1.2	Motivação	16
1.3	Definição do Problema	16
1.4	Objetivo Geral e Específicos	17
1.5	Estrutura do Trabalho	17
2	Referencial Teórico	18
2.1	Motilidade Celular	18
2.2	Segmentação	21
2.2.1	Erosão	22
2.2.2	Dilatação	22
2.2.3	Abertura	23
2.2.4	Fechamento	23
2.2.5	<i>Top-Hat</i>	23
2.2.6	Reconstrução por abertura	23
2.3	Rotulação	24
2.4	Rastreamento	25
2.5	Forma de Validação dos Algoritmos	26
2.5.1	Rotina de Segmentação	28
2.5.2	Rotina de Rastreamento	29
3	Trabalhos Relacionados	33
3.1	KTH-SE	33
3.2	LEID-NL	36
3.3	IMCB-SG	36
3.3.1	<i>Datasets 2D</i>	36
3.3.2	<i>Datasets 3D</i>	37
3.4	HD-Har-GE	38
3.5	CUNI-CZ	39
3.6	MU-CZ	40
3.7	FR-Ro-GE	41
3.8	UZH-CH	41
3.9	NOTT-UK	42
3.10	KIT-GE	43
3.11	FR-Be-GE	44

4	Resultados	46
4.1	Primeira Fase: Segmentação	46
4.1.1	Pós-processamento	48
4.1.2	<i>Dataset</i> Fluo-C3DL-MDA231	48
4.1.3	<i>Dataset</i> Fluo-C3DH-H157	49
4.1.4	<i>Dataset</i> Fluo-N3DH-SIM	50
4.1.5	<i>Dataset</i> Fluo-N3DH-CHO	51
4.1.6	<i>Dataset</i> Fluo-C2DL-MSK	52
4.1.7	<i>Dataset</i> Fluo-N2DH-GOWT1	53
4.1.8	<i>Dataset</i> PhC-C2DH-U373	54
4.2	Segunda Fase: Rotulação	54
4.3	Terceira Fase: Rastreamento	55
4.4	Resultados Obtidos e Discussão	59
4.4.1	Dificuldades	63
5	Conclusão	64
	Referências	66
	Anexos	69
	ANEXO A Descrição dos Datasets	70

1

Introdução

Este trabalho é uma continuação do trabalho de conclusão de curso (TCC) de [Sousa \(2015\)](#) voltado para o rastreamento celular. Neste capítulo será apresentada a importância de realizar o rastreamento de células e como a motilidade celular contribui para isto. Além disso, serão apresentados os principais desafios enfrentados para se realizar o rastreamento, como a Computação pode contribuir para este problema com seus algoritmos para realizar a segmentação, a rotulação e a vinculação — rastreamento — das células, e as contribuições deste trabalho em relação ao de [Sousa \(2015\)](#).

1.1 Contextualização

A motilidade celular está ligada a diversos processos biológicos e patológicos. A motilidade dirigida das células epiteliais, por exemplo, está ligada ao rápido fechamento de feridas. Em casos de câncer, em que a célula encontra-se em fase de metástase, a motilidade celular está ligada à disseminação da malignidade para outros locais do corpo ([LENZ, 2008](#)).

Levando isso em consideração, percebe-se que o estudo da motilidade celular tem um grande impacto na área médica, pois poder localizar uma célula, verificar para onde ela está se locomovendo e analisar o seu comportamento durante este trajeto podem ajudar na identificação e na cura de doenças.

De acordo com [Zimmer et al. \(2006\)](#), para se entender a forma e o porquê das células se moverem, são necessários conhecimentos advindos de outras áreas. Por exemplo, para se ter um entendimento completo dos movimentos celulares são necessários modelos mecânicos baseados nos princípios da Física, como também dos conhecimentos provenientes da Biologia sobre o comportamento das células com o meio em seu entorno.

Dessa forma, faz-se necessário buscar informações das diversas áreas do conhecimento que possam ter alguma relação ou que possam contribuir de alguma forma para o entendimento

da movimentação celular.

Segundo [Lenz \(2008\)](#), o rastreamento de uma célula pode ser afetado pelo seu tipo, pois cada uma possui características próprias tanto em relação à sua forma quanto ao seu modo de locomoção. A locomoção das células se dá por meio de alterações em seu citoesqueleto — estrutura responsável por todas as movimentações celulares, sejam elas intracelulares ou extracelulares. Essas alterações podem ser de forma regular ou irregular. Dependendo do local em que as células estão se movendo, os movimentos podem ser realizados de duas maneiras:

- a) Por meio do nado: quando o movimento é realizado em um ambiente líquido. Algumas células usam seus flagelos ou o batimento coordenado dos seus cílios para se impulsionarem pelo líquido circundante ([LENZ, 2008](#)).
- b) Por meio do rastejamento: quando o movimento é realizado em um ambiente de superfície rígida. Requer uma atividade coordenada do citoesqueleto, da membrana e do sistema de adesão da célula. A movimentação se dá pela projeção e retração da célula sobre o substrato ([LENZ, 2008](#)).

Devido a essas alterações na estrutura das células durante sua locomoção, um dos desafios relacionados ao rastreamento está ligado à identificação das mesmas, pois deve-se estar atento às mudanças que podem ocorrer tanto em seu tamanho quanto em sua forma, que acabam dificultando em encontrar um padrão para representar cada uma delas.

Outro desafio do rastreamento celular, e que é considerado por [Zimmer et al. \(2006\)](#) um problema recorrente ao rastrear múltiplas células, é quando duas células se tocam ou se sobrepõem umas às outras. Isso faz com que a identificação das fronteiras seja difícil de identificar automaticamente, assim como a olho nu.

Para que seja possível realizar esses estudos com relação à motilidade celular, é necessário que as informações da locomoção das células sejam armazenadas. Esse armazenamento é feito por meio de imagens, podendo estas serem 2D ou 3D. Dessa forma, além das características celulares é preciso estar atento também às características das imagens, pois ambas irão interferir no reconhecimento das células podendo dificultar sua segmentação, ou seja, em sua separação do restante da imagem.

Segundo [Zimmer et al. \(2006\)](#) a forma de aquisição das imagens também pode interferir no rastreamento celular, pois as modalidades de microscopia utilizadas para gerar as imagens podem influenciar no aparecimento das células nas mesmas. As modalidades não fluorescentes, como por exemplo, a de contraste de fase (PhC) e a de contraste de interferência diferencial (DIC), podem interferir na forma de exibição das células dependendo de suas posições em relação ao plano focal. As células podem aparecer com regiões mais escuras em suas bordas circundadas por regiões claras, ou vice-versa. Nesses casos, a intensidade média da região ocupada por uma

célula pode ser aproximadamente igual à do fundo local da imagem, dificultando o processo de segmentação.

Levando isso em consideração, tem-se aqui outro desafio do rastreamento celular, que é o desaparecimento momentâneo das células devido à interferência causada pelo método utilizado para a aquisição das imagens.

Segundo [Massoudi, Semenovich e Sowmya \(2012\)](#), para se ter um sistema de rastreamento bem-sucedido é necessário que todos os desafios comuns do rastreamento de células, incluindo a entrada e saída de novas células da tela e a divisão celular — mitose —, sejam satisfeitos. De acordo com [Zimmer et al. \(2006\)](#), o estudo do desenvolvimento de embriões, da proliferação de células-tronco ou de células cancerígenas possuem um interesse particular na detecção das divisões celulares. Isso se deve ao fato de permitir a construção de árvores de linhagem celular e a análise de características celulares em relação a seus ancestrais.

De acordo com ([CADART et al., 2014](#)), o objetivo da mitose é realizar a segregação precisa e simétrica da informação genética, na qual ocorre a formação dos *spindles* mitóticos e também a reorganização da estrutura celular que afetam praticamente todos os processos celulares. Durante a mitose, as células abandonam sua forma plana e achatada e se tornam arredondadas, gerando uma reorganização completa do citoesqueleto. Esse arredondamento é essencial pois gera o espaço geométrico necessário para a formação do *spindle* mitótico, que, por sua vez, é responsável pela simetria da divisão celular. Dependendo do local onde o *spindle* mitótico é formado durante a divisão celular, as células geradas podem ter tamanhos simétricos ou assimétricos. Além disso, essas alterações drásticas de forma durante a mitose não afetam apenas a célula que está sendo dividida, mas também as suas vizinhas.

Diferente das outras características celulares, a motilidade de uma célula geralmente é caracterizada por mais de uma imagem. Segundo [Zimmer et al. \(2006\)](#), isso se deve ao fato de que para se quantificar de forma confiável as características de movimento de células individuais são necessários muitos pontos de tempo. Além disso, a variabilidade intercelular acarreta que um grande número de células deve ser observado. Por isso, devido à quantidade de imagens necessárias para armazenar os dados do processo de movimentação das células, o trabalho de analisar e fazer a trajetória de cada célula manualmente se torna penoso e de difícil reprodução, principalmente em imagens 3D. Isso acaba implicando na necessidade do desenvolvimento de algoritmos computacionais que possam realizar esse trabalho de forma automática e prática.

Dessa forma, os algoritmos computacionais devem ser implementados visando solucionar os desafios inerentes ao rastreamento celular que são: a identificação das células; o desaparecimento momentâneo; células que se tocam ou se sobrepõem; entrada e saída de novas células; e divisão celular.

A fim de contribuir neste contexto, a Computação disponibiliza diversos meios para o desenvolvimento e implementação de um algoritmo de rastreamento como por exemplo: métodos

de segmentação para que todas células possam ser separadas do restante da imagem; e métodos de rotulação para que cada uma delas possam ser identificadas de forma única. Porém, para que isso seja possível, faz-se necessário adquirir os dados das células para estudá-los.

Desde a sua criação em 2013, o “*Cell Tracking Challenge*” ([SOLORZANO, 2017a](#)) vem contribuindo de forma significativa para os estudos do rastreamento de células. Além dos *datasets*, que antes de sua criação eram privados e em pouca quantidade, ele disponibiliza também um *software* para realizar a avaliação dos métodos implementados, criando uma forma padrão de avaliação dos resultados que até então não existia. Isso facilitou o acesso aos dados necessários tanto para a criação dos algoritmos computacionais quanto para a validação dos mesmos.

1.2 Motivação

Estudos feitos pelo Instituto Nacional de Câncer José Alencar Gomes da Silva (INCA) mostram que 63% dos óbitos que ocorreram em 2008 em todo o mundo foram em consequência das doenças e agravos não transmissíveis (DANT), sendo 21% destes causados pelo câncer. Esses estudos mostraram também que em 2012 ocorreram 14,1 milhões de novos casos de câncer e 8,2 milhões de óbitos em todo o mundo. Estima-se também que, para o Brasil, em 2018 e 2019, haja uma ocorrência de 600 mil novos casos de câncer para cada ano ([INCA, 2017](#)).

O estudo da motilidade celular está relacionado a diversos fatores de suma importância como a cura e prevenção de doenças, inclusive do câncer. A Computação contribui neste contexto com a criação de algoritmos para auxiliar na análise das imagens, diminuindo o esforço necessário para realizar esta tarefa, além de minimizar os erros que podem ser ocasionados pela análise de forma manual.

1.3 Definição do Problema

Como dito anteriormente, analisar a trajetória de uma célula durante seu desenvolvimento de forma manual é um trabalho árduo e que pode levar a erros. Para ajudar nisso, necessita-se de algoritmos que possam fazer esse processo de forma automática. Porém, criar um algoritmo universal que possa realizar o rastreamento para todos os tipos de células e em todos os tipos de imagem não é tão fácil. Segundo [Solorzano \(2017d\)](#), isso se deve a 2 fatores:

- a) As modalidades de microscopia: como visto anteriormente, elas podem influenciar no aparecimento das células nas imagens, dificultando durante o processo de segmentação.
- b) A natureza variável dos *datasets*, como por exemplo: as dimensões das imagens (2D ou 3D); o nível de ruído; a qualidade da imagem em geral; os diferentes tipos de células; e se o objeto de estudo é a célula por completo ou apenas seu núcleo.

Levando isso em consideração, neste trabalho pretendeu-se criar um algoritmo de rastreamento que pudesse identificar a divisão celular e que pudesse ser aplicado ao maior número possível de *datasets* dentre os 11 disponíveis (ver Anexo A).

1.4 Objetivo Geral e Específicos

O objetivo geral deste trabalho consiste em implementar um método de rastreamento que possa detectar as divisões celulares e a entrada e saída de novas células. Dessa forma, pretende-se melhorar o método de rastreamento utilizado no TCC de Sousa (2015), que até então não possuía suporte a esses casos.

Outras contribuições deste trabalho, além das já citadas anteriormente, são: a alteração do método de rotulação utilizado por Sousa (2015) que possuía limitações quando aplicado a imagens com uma grande quantidade de componentes conexas; e a adição de novos *datasets*, pois em Sousa (2015) apenas o *dataset* Fluo-C3DL-MDA231 tinha sido utilizado.

Dessa forma, têm-se como objetivos específicos:

- a) Implementação de novas formas de segmentação: devido à adição dos novos *datasets*, é necessário encontrar os métodos de segmentação mais adequados para cada um, visto que cada *dataset* possui suas peculiaridades. Além disso, a fim de melhorar os resultados para o *dataset* Fluo-C3DL-MDA231, novas formas de segmentação diferentes da utilizada por Sousa (2015) foram testadas.
- b) Implementação do novo método de rotulação: devido à limitação da abordagem de *label equivalence* utilizada por Sousa (2015), optou-se por alterar o método de rotulação para o de *searching and label propagation*. Pretende-se com essa nova forma de rotulação eliminar a limitação do método antigo de forma que a quantidade de rótulos disponíveis não esgote antes de concluir a identificação de todas as componentes conexas.
- c) Implementação do método para detecção da divisão celular.
- d) Implementação dos ajustes necessários para detecção da entrada e saída de novas células.
- e) Validação do algoritmo implementado.

1.5 Estrutura do Trabalho

Este trabalho está estruturado da seguinte maneira: no Capítulo 2, são apresentados os trabalhos relacionados; no Capítulo 3, é apresentado o referencial teórico sobre os principais passos necessários para a implementação do algoritmo proposto; no Capítulo 4, são apresentados o algoritmo resultante, assim como os resultados de sua validação e as dificuldades encontradas; e por fim, no Capítulo 5, são apresentadas as conclusões.

2

Referencial Teórico

Esse trabalho é composto pelo desenvolvimento do algoritmo proposto e sua validação, sendo que a fase de desenvolvimento se divide em 3 fases principais: segmentação, rotulação e rastreamento. Nesse capítulo será apresentado um pequeno resumo em relação a cada uma destas fases identificando alguns dos tipos de segmentação existentes, os principais métodos de rotulação e rastreamento utilizados, assim como a forma de validação dos resultados. Além disso, será apresentado também um pequeno resumo sobre o funcionamento da motilidade celular.

2.1 Motilidade Celular

Como dito anteriormente, a motilidade celular está ligada ao modo de movimentação da célula que varia dependendo do tipo e forma da mesma. Por exemplo, os fibroblastos possuem um movimento mais lento e descontínuo e sua forma tende a ficar triangular durante a movimentação. Os glóbulos brancos possuem um movimento mais rápido com formação de várias protuberâncias na parte frontal, onde a maior protuberância determina a direção do movimento. Os ceratócitos epiteliais de peixes possuem um movimento contínuo e deslizante, mantendo uma forma simples semicircular. Diferente dos outros tipos de células em que os movimentos são descritos como uma caminhada aleatória, os ceratócitos podem manter sua forma e a direção do movimento por vários minutos de cada vez (LEE, 2018)

Devido a essa variabilidade na forma e no modo de movimento existente entre os diferentes tipos de células, o processo de derivar um mecanismo básico subjacente à motilidade celular acaba sendo dificultado. Além disso, os modelos de motilidade variam em seu grau de complexidade, abordagem e aspecto do movimento sendo modelado (LEE, 2018).

Segundo Lee (2018), existem duas categorias amplas de modelos matemáticos:

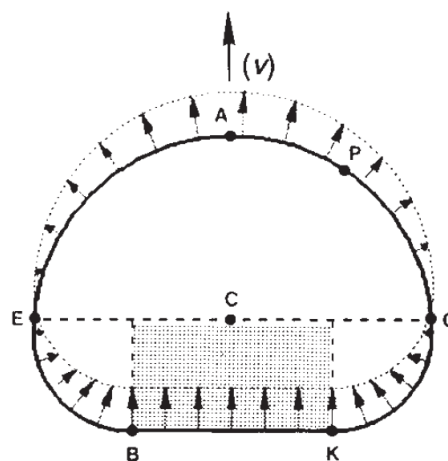
- a) *Top-down*: que se concentra na forma e no movimento da célula inteira.

b) *Bottom-up*: que se concentra nos mecanismos moleculares subjacentes à motilidade celular.

A escolha do ceratócito epitelial dos peixes para a modelagem matemática é devido a sua forma simples e o modo de movimento de deslizamento facilitarem a ligação de processos de escala molecular ao formato e à velocidade da célula (LEE, 2018). Além disso, segundo Vallotton et al. (2005), com a descoberta do fluxo retrógrado nos ceratócitos, foi possível destacar a similaridade deles com outras células de rastreamento, validando o seu uso como um modelo geral para a motilidade celular.

Um dos modelos matemáticos mais simples é o modelo de extensão radial graduada (*Graded Radial Extension - GRE*). Ele define a magnitude e a direção da extensão lamelar exigida para manter a forma semicircular e a velocidade constante da célula em movimento. Segundo Lee et al. (1993), a extensão e retração dos ceratócitos ocorrem de forma perpendicular à borda da célula, e os graus de distribuição graduada da extensão e retração ao longo da margem da célula são responsáveis por manter a forma e o tamanho da célula durante a locomoção. Os pontos E e Q (Figura 1) são chamados de ponto de equilíbrio, pois representam as regiões em que não ocorrem extensão nem retração. Uma linha imaginária desenhada entre esses dois pontos representa a divisão entre a região com extensão — representada pelo semicírculo EAQ (Figura 1) — e a região com retração — representada pelo semicírculo EBKQ (Figura 1). Esse arranjo simétrico e graduado dos graus de extensão e retração perpendiculares à borda da célula resultam na translocação direta da célula assim como da manutenção da sua forma e do seu tamanho.

Figura 1 – Modelo idealizado da forma de um ceratócito se movendo de acordo com o modelo GRE



Fonte – Lee et al. (1993).

O modelo GRE prediz que as regiões de extensão e retração das bordas se movem ao longo da circunferência da célula durante a locomoção (Figura 2). Por exemplo, a região P do lado direito da extensão da borda AQ se move ao longo da circunferência na direção horária. Similarmente, regiões do lado esquerdo da extensão da borda AE se movem ao longo da circunferência na direção anti-horária.

$$x' = \frac{ms + \sqrt{(R^2(1+m^2) - s^2)}}{1+m^2} \quad (2.3)$$

$$y' = \frac{m^2s + m\sqrt{(R^2(1+m^2) - s^2)}}{1+m^2} \quad (2.4)$$

$$\theta = 90 - \arctan\left(\frac{y' - s}{x'}\right) \quad (2.5)$$

Apesar do modelo GRE descrever o movimento cinemático do ceratócito, ele não faz nenhuma suposição sobre os mecanismos moleculares envolvidos. Outros modelos podem ser encontrados em [Ziebert e Aranson \(2016\)](#).

2.2 Segmentação

O processo de segmentação tem como papel facilitar a análise das imagens. Ele é responsável por agrupar os pixels com características em comum formando componentes conexas homogêneas. Por esse motivo, é considerado por [Dass, Priyanka e Devi \(2012\)](#) um dos primeiros passos a ser realizado ao se trabalhar com a análise de imagens.

O critério de segmentação para uma determinada imagem depende da forma de obtenção da mesma, o que faz com que este critério varie dependendo da imagem a ser analisada. Além disso, é necessário que após a segmentação o *foreground* — objeto a ser estudado — e o *background* possam ser facilmente identificados.

Segundo [Khan e Ravi \(2013\)](#), cada método de segmentação possui suas próprias vantagens e desvantagens e podem ser divididos em categorias baseadas na abordagem utilizada no seu desenvolvimento. Alguns grupos que podem ser citados são:

- a) Métodos baseados na intensidade: classificam a imagem em duas classes. Os pixels pertencentes a uma determinada faixa de valores representa uma classe e o restante dos pixels na imagem representam a outra classe. Um exemplo desse grupo são os métodos que utilizam o *threshold*.
- b) Métodos baseados na descontinuidade: baseiam-se no princípio de que há uma variação de intensidade entre os pixels, o que acabam gerando bordas entre dois objetos distintos. As bordas são definidas como mudanças abruptas nos níveis de intensidade entre pixels vizinhos em uma determinada direção, resultando em uma descontinuidade. Um exemplo desse grupo é o operador de Sobel.
- c) Métodos baseados na região: partem do princípio da homogeneidade. Considera que as intensidades dos pixels dentro de uma mesma região possuem características similares e são diferentes dos pixels de outras regiões.

- d) Métodos baseados no agrupamento: o objetivo é encontrar grupos nos dados. Um *cluster* normalmente contém um grupo de pixels similares que pertencem a uma determinada região e diferem de outras regiões. Um exemplo desse grupo é o algoritmo de *K-means*. Os métodos baseados na intensidade estão englobados nesse grupo.

Em [Smochina, Herghelegiu e Manta \(2011\)](#) é possível encontrar algumas técnicas de processamento de imagem utilizadas na segmentação de imagens microscópicas.

Como pode ser visto, não existe um método padrão e universal de segmentação que possa ser aplicado a todos os tipos de imagem. Cada método possui suas próprias características que o torna melhor para determinada situação.

Nas seções seguintes serão apresentados alguns dos métodos utilizados para a composição dos algoritmos de segmentação implementados durante a execução deste trabalho.

2.2.1 Erosão

Um dos usos mais simples da erosão é para eliminar detalhes irrelevantes da imagem. A erosão de uma imagem f por um elemento estruturante b em qualquer posição (x, y) é definida como o valor *mínimo* da imagem na região coincidente com b quando a origem de b está em (x, y) . Portanto, a erosão em (x, y) de uma imagem f por um elemento estruturante b é dada pela equação 2.6:

$$[f \ominus b](x, y) = \min_{(s, t) \in b} \{f(x + s, y + t)\} \quad (2.6)$$

onde, x e y são coordenadas da imagem f , e s e t são coordenadas do elemento estruturante b . Um elemento estruturante é uma matriz que identifica o pixel na imagem sendo processada e define a vizinhança usada no processamento de cada pixel. Ele pode ser 2D ou 3D, com coordenadas (x, y) e (x, y, z) , respectivamente, em que: x representa a largura; y representa a altura; e z representa a profundidade.

2.2.2 Dilatação

Um dos usos mais simples da dilatação é para juntar lacunas. A dilatação de uma imagem f por um elemento estruturante b em qualquer posição (x, y) é definida como o valor *máximo* da imagem na região coincidente com b quando a origem de b está em (x, y) . Portanto, a dilatação em (x, y) de uma imagem f por um elemento estruturante b é dada pela equação 2.7:

$$[f \oplus b](x, y) = \max_{(s, t) \in b} \{f(x - s, y - t)\} \quad (2.7)$$

onde, x e y são coordenadas da imagem f , e s e t são coordenadas do elemento estruturante b .

2.2.3 Abertura

A abertura geralmente suaviza o contorno de um objeto e elimina protruções finas. A abertura de uma imagem f por um elemento estruturante b é definida como a erosão de f por b seguida por uma dilatação do resultado por b , e é dada pela fórmula 2.8.

$$f \circ b = (f \ominus b) \oplus b \quad (2.8)$$

2.2.4 Fechamento

O fechamento geralmente elimina pequenos buracos e preenche pequenas lacunas no contorno. O fechamento de uma imagem f por um elemento estruturante b é definido como a dilatação de f por b seguida por uma erosão do resultado com b , e é dado pela fórmula 2.9.

$$f \bullet b = (f \oplus b) \ominus b \quad (2.9)$$

2.2.5 Top-Hat

O *top-hat* é útil para aprimorar os detalhes de uma imagem na presença de sombreamento. Dessa forma, um uso importante dele é na correção dos efeitos em uma imagem com iluminação não-uniforme. O *top-hat* de uma imagem f é definido como f menos sua abertura, e é dado pela fórmula 2.10.

$$T_{hat}(f) = f - (f \circ b) \quad (2.10)$$

2.2.6 Reconstrução por abertura

A reconstrução por abertura é usada para remover objetos, com intensidades de pixel claras, menores que o elemento estruturante. O primeiro passo é aplicar uma erosão na imagem de entrada para gerar a imagem marcadora.

A reconstrução por abertura de uma imagem f é definida como a reconstrução por dilatação da erosão aplicada a esta imagem e é dada pela fórmula 2.11. O objetivo dela é preservar a forma dos componentes da imagem que permanecem após a erosão.

$$O_R(f) = R_f^D[(f \ominus b)], \text{ onde } R_f^D \text{ denota a operação de reconstrução por dilatação.} \quad (2.11)$$

A reconstrução por dilatação de uma imagem g por uma imagem marcadora f é definida como a dilatação geodésica de f com respeito a g repetida até alcançar a estabilidade (imagem na iteração $k+1$ igual à da iteração k) e é dada pela fórmula 2.12.

$$R_g^D(f) = D_g^{(k)}(f), \text{ com } k \text{ de modo que } D_g^{(k)}(f) = D_g^{(k+1)}(f). \quad (2.12)$$

Sejam f e g as imagens marcadora e máscara, ambas de mesmo tamanho e que $f \leq g$, ou seja, o domínio de f é um subconjunto do domínio de g e para qualquer valor x no domínio de f , que deve ser o mesmo domínio de ambos, $f(x) \leq g(x)$. A dilatação geodésica de tamanho 1 é dada pela fórmula 2.13.

$$D_{(g)}^{(1)}(f) = (f \oplus b) \wedge g, \text{ onde } \wedge \text{ denota o operador pontual de mínimo.} \quad (2.13)$$

Então, a dilatação geodésica de tamanho 1 é obtida pela dilatação de f pelo elemento estruturante b e então seleciona o mínimo entre o resultado e g para cada ponto (x, y) . A dilatação geodésica de tamanho n de f com respeito a g é dada pela fórmula 2.14.

$$D_{(g)}^{(n)}(f) = D_{(g)}^{(1)}(f)[D_{(g)}^{(n-1)}(f)], \text{ onde } D_{(g)}^{(0)}(f) = f. \quad (2.14)$$

2.3 Rotulação

A rotulação tem como papel atribuir rótulos distintos a cada componente conexa de uma imagem permitindo distingui-las umas das outras. Uma componente conexa representa um conjunto de pixels em que todos os pixels estão interligados entre si e seus valores de intensidade pertencem a uma mesma faixa de valores. Um pixel está interligado a outro quando são adjacentes. Pixels que estão distantes um do outro pertencerão à mesma componente conexa desde que haja pixels adjacentes que interliguem eles. De acordo com os valores, os pixels podem pertencer a dois grupos: *foreground*, quando os valores dos pixels são diferentes de 0; e *background*, quando os valores dos pixels são iguais a 0.

De acordo com He, Chao e Suzuki (2010), os algoritmos de rotulação podem ser classificados em dois tipos:

- a) Algoritmos de *label equivalence*: nesse tipo de abordagem é necessário que a imagem seja lida no mínimo duas vezes. Na primeira leitura, é designado um rótulo temporário para cada pixel das componentes conexas, podendo uma mesma componente ter mais de um rótulo. Na segunda leitura os rótulos de cada componente conexa são unificados, de modo que um dos rótulos temporários da componente seja selecionado para representar esta componente e utilizado para substituir todos os outros rótulos da mesma.
- b) Algoritmos de *searching and label propagation*: nesta abordagem, diferente da anterior, a ideia é que cada componente conexa receba o seu rótulo definitivo em uma única leitura. Ao encontrar um pixel de uma componente conexa, o mesmo recebe um valor de rótulo

e este é propagado para os pixels adjacentes a ele. Segundo [Sousa \(2015\)](#), embora uma única leitura seja necessária, a forma de processar a imagem é irregular, dependendo das formas das componentes conexas da imagem.

2.4 Rastreamento

O rastreamento tem como objetivo estimar a trajetória de um objeto no plano da imagem à medida que ele se move em torno de uma cena localizando sua posição em cada imagem do vídeo. Isso é feito atribuindo rótulos consistentes aos objetos rastreados em cada imagem do vídeo de modo a fornecer a região completa na imagem que é ocupada pelo objeto a cada instante. Além disso, dependendo do domínio de rastreamento, outras informações do objeto podem ser adquiridas com o rastreamento como, por exemplo, sua orientação, sua área e sua forma ([YILMAZ; JAVED; SHAH, 2006](#)).

Segundo [Yilmaz, Javed e Shah \(2006\)](#), existem três passos principais para que seja realizada a análise de um vídeo: detectar os objetos de interesse; rastrear esses objetos de uma imagem a outra; e analisar o trajeto feito pelo objeto para reconhecer seu comportamento. Porém, dificuldades durante o rastreamento podem surgir devido ao movimento abrupto do objeto, mudanças de aparência tanto do objeto quanto da cena, oclusões do objeto, movimento da câmera e mudanças de iluminação. De acordo com [Sonka, Hlavac e Boyle \(2008\)](#), um conhecimento prévio sobre o movimento da câmera, e informações sobre o intervalo de tempo entre imagens consecutivas podem ajudar a escolher uma técnica de análise de movimento adequada, embora não haja um algoritmo geral para isso.

De uma forma geral, o rastreamento necessita da localização e / ou a forma do objeto em cada imagem do vídeo. De acordo com [Yilmaz, Javed e Shah \(2006\)](#), os objetos podem ser definidos como qualquer coisa que seja de interesse para análise posterior e podem ser representados por suas formas e aparências como, por exemplo, imagens de barcos no mar em uma imagem em que o fundo representa o mar, imagens de peixes dentro de um aquário em uma imagem em que o fundo representa o aquário, entre outros. As características utilizadas para o rastreamento estão intimamente relacionadas à representação do objeto levando em consideração suas peculiaridades. Isso é realizado dessa forma para que os objetos possam ser facilmente distinguidos na cena.

Os métodos de rastreamento necessitam de um mecanismo para a detecção dos objetos, seja essa detecção feita em cada imagem do vídeo ou apenas quando o objeto aparece pela primeira vez no vídeo. Segundo [Yilmaz, Javed e Shah \(2006\)](#), uma abordagem comum é detectar o objeto utilizando apenas as informações de um único quadro, porém, alguns métodos fazem uso das informações temporais calculadas a partir de uma sequência de imagens para reduzir o número de falsas detecções. Dessa forma, dadas as regiões dos objetos nas imagens, o rastreador realiza a correspondência entre os objetos de uma imagem com os objetos correspondentes da

imagem seguinte.

Segundo [Maška et al. \(2014\)](#), os passos de detectar os objetos (fase de segmentação) e estabelecer suas correspondências entre as imagens (fase de rastreamento) podem ser realizados de forma separada (abordagem de *tracking by detection*) ou em conjunto (abordagem de *tracking by evolution*).

Na abordagem de *tracking by detection*, as regiões dos objetos em cada imagem são obtidas por meio de um algoritmo de detecção de objetos e, em seguida, o rastreador faz a correspondência entre os objetos nas imagens. Na abordagem de *tracking by evolution*, a região do objeto e suas correspondências são estimadas de forma conjunta por meio da atualização iterativa da localização do objeto e das informações das regiões obtidas nas imagens anteriores.

De acordo com [Meijering, Dzyubachyk e Smal \(2012\)](#), a forma mais simples de resolver o problema da associação é associar o objeto segmentado em qualquer imagem ao objeto mais próximo na imagem seguinte, sendo que essa relação de proximidade pode se referir à distância espacial, à diferença de intensidade, ao volume do objeto, entre outras características. Porém, para que essa solução do vizinho mais próximo funcione bem é necessário que as células estejam bem separadas em pelo menos uma das dimensões do espaço de características.

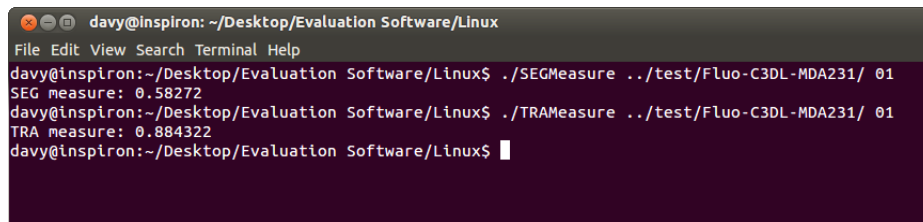
2.5 Forma de Validação dos Algoritmos

Para a validação dos algoritmos de segmentação e de rastreamento, foi utilizado o “*Evaluation Software*” ([SOLORZANO, 2017b](#)), que possui duas rotinas de avaliação, sendo uma para a fase de segmentação e outra para a fase de rastreamento. A escolha pelo uso desse *software* se deu pelo fato dele realizar a comparação dos resultados obtidos pelos algoritmos apresentados com os *ground truth* — as respostas corretas — de forma automática, além de gerar um arquivo para cada avaliação contendo informações adicionais. Mais detalhes do funcionamento de cada rotina podem ser vistos nas seções [2.5.1](#) e [2.5.2](#). Outro motivo para o uso desse *software* foi o fato de permitir a comparação dos resultados obtidos pelos algoritmos apresentados neste trabalho com os algoritmos utilizados por outros pesquisadores, visto que estes mesmos pesquisadores utilizaram esse *software* para validar seus algoritmos.

A execução das rotinas se dá por meio do terminal de comando. No terminal é feita a chamada da rotina a ser executada passando-se como parâmetros o diretório do *dataset* e o número do vídeo a ser analisado. O resultado da rotina é exibido logo em seguida no próprio terminal, e um arquivo para cada rotina (SEG_log.txt ou TRA_log.txt) é gerado. Um exemplo de execução das rotinas, tanto de segmentação quanto de rastreamento, pode ser visto na Figura [3](#). Para que os resultados possam ser avaliados, é necessário que os mesmos estejam no formato especificado no arquivo “*Naming and file content conventions.pdf*” ([SOLORZANO, 2017c](#)).

A avaliação se dá pela comparação dos resultados obtidos pelo algoritmo desenvolvido

Figura 3 – Execução das Rotinas de Avaliação da Segmentação e do Rastreamento

A terminal window titled 'davy@inspiron: ~/Desktop/Evaluation Software/Linux'. The window shows the execution of two commands: './SEGMeasure ../test/Fluo-C3DL-MDA231/ 01' which outputs 'SEG measure: 0.58272', and './TRAMeasure ../test/Fluo-C3DL-MDA231/ 01' which outputs 'TRA measure: 0.884322'. The prompt returns to 'davy@inspiron:~/Desktop/Evaluation Software/Linux\$' after each command.

```
davy@inspiron: ~/Desktop/Evaluation Software/Linux
File Edit View Search Terminal Help
davy@inspiron:~/Desktop/Evaluation Software/Linux$ ./SEGMeasure ../test/Fluo-C3DL-MDA231/ 01
SEG measure: 0.58272
davy@inspiron:~/Desktop/Evaluation Software/Linux$ ./TRAMeasure ../test/Fluo-C3DL-MDA231/ 01
TRA measure: 0.884322
davy@inspiron:~/Desktop/Evaluation Software/Linux$
```

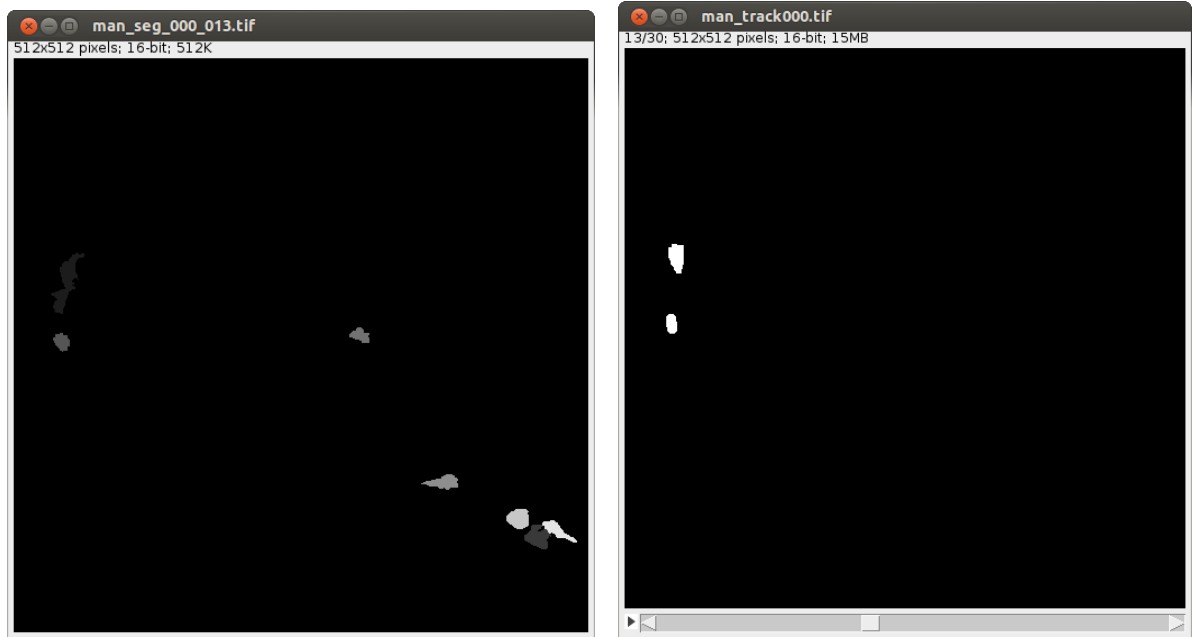
Fonte – O autor, 2018.

com os resultados dos *ground truth*. Segundo Sousa (2015), os *ground truth* foram criados pelos especialistas responsáveis pelo “Cell Tracking Challenge”, e cada fase — segmentação e rastreamento — possui seus próprios *ground truth*.

Diferente da avaliação realizada para o rastreamento, a fase de segmentação não utiliza todas as imagens do vídeo, nem todos os *frames* — no caso das imagens 3D. Por exemplo, o *dataset* Fluo-C3DL-MDA231 possui 12 imagens com 30 *frames*, porém, para a avaliação da segmentação, apenas alguns *frames* de algumas imagens são selecionados para fazer a verificação. Isso pode afetar a avaliação da segmentação pois apenas uma parte das informações originais são utilizadas para a validação do conjunto inteiro.

Tomando como exemplo uma imagem 3D com 30 *frames* em que uma célula ocupa os *frames* de 1 a 7 desta imagem, porém, na imagem correspondente a esta do *ground truth*, apenas o *frame* 7 foi utilizado para a avaliação, e no resultado obtido pelo algoritmo de segmentação esta mesma célula foi identificada apenas nos *frames* de 3 a 6. Por meio deste exemplo, é possível perceber que mesmo tendo identificado a célula em questão, ela não seria considerada pelo software de avaliação, visto que a mesma não está presente no resultado obtido pela segmentação no *frame* utilizado pelo software — o *frame* 7.

Outra diferença que pode ser notada por meio da Figura 4 é em relação ao conteúdo presente na imagem do *ground truth* para a fase de segmentação e de rastreamento. O mesmo *frame*, para a mesma imagem, contém informações diferentes para a avaliação da fase de segmentação e de rastreamento. Na imagem referente ao *ground truth* da segmentação é apresentada uma quantidade maior de células que a imagem referente ao *ground truth* do rastreamento no mesmo *frame*. Dessa forma, pode acontecer de, mesmo não obtendo um valor de acurácia satisfatório para a fase de segmentação, obter um valor satisfatório para a fase de rastreamento, visto que para o rastreamento não é necessário que a célula seja identificada em todos os *frames* da imagem.

Figura 4 – *Ground Truth* do Dataset Fluo-C3DL-MDA231(a) *Ground truth* da segmentação: imagem “man_seg_000_13”.(b) *Ground truth* do rastreamento: imagem do frame 13 de “man_track000”.

2.5.1 Rotina de Segmentação

A rotina de segmentação (SEG) tem como objetivo medir o grau de coincidência entre as regiões das células segmentadas e as regiões das células reais, e é baseada nos índices de Jaccard. O índice de Jaccard (J) de um dado objeto (i) é dado pela Fórmula 2.15:

$$J_i(S_i, R_j) = \frac{|R_j \cap S_i|}{|R_j \cup S_i|} \quad (2.15)$$

na qual, S_i representa o conjunto de pixels do objeto obtido pelo algoritmo de segmentação; e R_j representa o conjunto de pixels do objeto correspondente no *ground truth*. Os objetos só serão considerados correspondentes se e somente se $|R_j \cap S_i| > 0,5 \cdot |R_j|$, ou seja, mais de 50% da célula do *ground truth* deve estar incluso na célula segmentada pelo algoritmo apresentado.

Para cada célula do *ground truth* só pode haver uma única célula correspondente dentre as obtidas pelo algoritmo de segmentação utilizado. Se não houver sobreposição significativa entre a célula do *ground truth* e a célula segmentada, o conjunto correspondente é definido como o conjunto vazio.

Os valores dos índices de Jaccard sempre estão no intervalo $[0, 1]$, no qual 1 significa correspondência perfeita e 0 significa que não houve correspondência.

A acurácia para um vídeo específico é calculada por meio do cálculo da média dos índices de Jaccard de todos os objetos presentes no *ground truth* do respectivo vídeo e é dada pela fórmula 2.16.

$$SEG = \frac{\sum_{i=1}^n J_i}{n} \quad (2.16)$$

Ao final, é criado um arquivo — SEG_log.txt — onde é possível ver quantas células existem em cada imagem do *ground truth*, o grau de correspondência entre cada uma dessas células e as identificadas pelo algoritmo de segmentação, além da média final da segmentação.

2.5.2 Rotina de Rastreamento

A rotina de rastreamento (TRA) tem como objetivo avaliar a habilidade dos algoritmos de rastreamento em detectar as células e segui-las no tempo.

O resultado do rastreamento de células é representado por um grafo orientado acíclico, cujos nós do grafo representam as células detectadas, e as arestas representam o vínculo entre elas no tempo. Existem duas semânticas para as arestas:

- a) *Track links*: são os casos em que não ocorre divisão celular e as células permanecem dentro do FoI. Nesse caso, o rótulo da célula permanece o mesmo em todas as imagens em que ela aparecer.
- b) *Parent links*: são os casos em que ocorre divisão celular. Nesse caso, são atribuídos novos rótulos, um para cada célula resultante da divisão celular, que irão identificá-las no restante das imagens.

A medida TRA calcula a distância entre o grafo gerado pelo algoritmo de rastreamento e do *ground truth*, ou seja, calcula a diferença entre os grafos a fim de medir a dificuldade para tornar o grafo gerado pelo algoritmo idêntico ao grafo do *ground truth*. Essa dificuldade é medida de forma automática por meio da quantidade de operações necessárias para tornar os grafos idênticos. As operações disponíveis, assim como os pesos de cada operação, podem ser vistas na Tabela 1. Os valores dos pesos de cada operação são de acordo com o esforço necessário para executar a operação correspondente de forma manual.

Para que a transformação ideal do grafo obtido pelo algoritmo de rastreamento no grafo do *ground truth* seja obtida, é necessário determinar as correspondências entre os nós de ambos os grafos. As correspondências são realizadas através da Fórmula 2.17:

$$|R_i^t \cap S_j^t| > 0.5 \cdot |R_i^t| \quad (2.17)$$

na qual t é a imagem, e i e j são os rótulos dos nós. Isto significa que mais de 50% de um nó específico i do *ground truth* (R) deve estar incluso no nó j do resultado obtido pelo algoritmo (S) na imagem t .

Após identificar cada nó, eles são classificados em quatro categorias:

Tabela 1 – Correspondência entre operações e pesos.

Operação	Peso	Descrição
Deletar nó	$w_{FP} = 1$	Requer um clique do mouse
Dividir nó	$w_{NS} = 5$	Requer desenhar uma divisão
Adicionar nó	$w_{FN} = 10$	Requer adicionar uma máscara completa
Deletar aresta	$w_{ED_1} = 0$ $w_{ED_2} = 1$	Requer um clique do mouse
Adicionar aresta	$W_{EA} = 1,5$	Mais difícil que deletar uma aresta pois requer determinar ambos os nós da aresta
Mudar semântica da aresta	$W_{EC} = 1$	Requer um clique do mouse

Fonte – Adaptado de Sousa (2015).

1. Falsos negativos (FN): são os nós do *ground truth* que não possuem nenhuma correspondência com os nós obtidos pelo algoritmo.
2. Falsos positivos (FP): são os nós obtidos pelo algoritmo que não possuem nenhuma correspondência com os nós do *ground truth*.
3. Verdadeiros positivos (TP): são os nós do *ground truth* que possuem correspondência com algum dos nós obtido pelo algoritmo.
4. Nós não divididos (NS): são os nós obtidos pelo algoritmo que possuem correspondência com mais de um dos nós do *ground truth*.

Após classificar cada nó, o procedimento calcula automaticamente quantas arestas devem ser removidas do grafo gerado pelo algoritmo de rastreamento (ED) por meio da Fórmula 2.18. As arestas podem ser de dois tipos:

- a) As arestas estão conectadas a um nó falso positivo (ED_1).
- b) As arestas conectam dois nós detectados corretamente, mas que não são vinculados no grafo do *ground truth* (ED_2).

$$ED = ED_1 + ED_2 \quad (2.18)$$

Assim como a rotina calcula quantas arestas precisam ser retiradas, ela também calcula quantas arestas estão faltando no grafo obtido pelo algoritmo de rastreamento e precisam ser adicionadas. Estas são as arestas do *ground truth* que não possuem nenhuma aresta correspondente no grafo obtido pelo algoritmo (EA).

Em seguida, o número de arestas que diferem na semântica são contabilizados (EC). O cálculo do *EC* é feito verificando todas as arestas que conectam os nós verdadeiros positivos de forma simultânea em ambos os grafos. Se a semântica da aresta do grafo do *ground truth* for diferente da semântica da aresta do grafo obtido pelo algoritmo de rastreamento, o contador é incrementado.

A transformação ideal para tornar o grafo obtido pelo algoritmo de rastreamento idêntico ao grafo do *ground truth* é realizado da seguinte maneira:

1. Todos os nós não divididos são separados — requer *NS* operações de dividir nó.
2. Todos os nós falsos negativos são adicionados — requer *FN* operações de adicionar nó.
3. Todos os nós falsos positivos são removidos — requer *FP* operações de deletar nó.
4. Tendo unificado os conjuntos de nós, são contadas quantas arestas precisam ser deletadas — requer *ED* operações de deletar aresta —, e quantas precisam ser adicionadas — requer *EA* operações de adicionar aresta.
5. Por fim, as arestas com semântica errada são corrigidas — requer *EC* operações de mudar semântica da aresta.

A distância entre os grafos então é considerada como a soma ponderada do número de operações necessárias para transformar o grafo obtido pelo algoritmo de rastreamento no grafo do *ground truth*, e é dada pela Fórmula 2.19.

$$TRA_P = w_{NS} \cdot NS + w_{FN} \cdot FN + w_{FP} \cdot FP + w_{ED_1} \cdot ED_1 + w_{ED_2} \cdot ED_2 + w_{EA} \cdot EA + w_{EC} \cdot EC \quad (2.19)$$

Se os grafos forem idênticos, o valor de TRA_P é igual a zero. O valor de TRA é normalizado no intervalo $[0, 1]$, no qual 1 significa correspondência perfeita, e 0 que não há correspondência. A acurácia final do rastreamento TRA é dada pela Fórmula 2.20:

$$TRA = 1 - \frac{\min(TRA_P, TRA_E)}{TRA_E} \quad (2.20)$$

na qual, TRA_E é o custo de criar um grafo idêntico ao grafo do *ground truth* a partir do zero, ou seja, é o TRA_P para resultados de rastreamento vazios. O operador de mínimo (\min) no numerador

é utilizado para prevenir resultados com valores negativos de *TRA* quando um grafo é computado totalmente errado, ou seja, quando é mais barato criar um grafo idêntico ao grafo do *ground truth* totalmente do zero do que transformar o grafo gerado pelo algoritmo de rastreamento idêntico ao do *ground truth*.

Ao final, é criado um arquivo — *TRA_log.txt* — onde podem ser vistas quantas operações foram realizadas, e em quais nós, no grafo gerado pelo algoritmo de rastreamento para torná-lo idêntico ao *ground truth*, além da média final do rastreamento.

3

Trabalhos Relacionados

Neste capítulo serão apresentados alguns trabalhos relacionados ao rastreamento de células. Os critérios utilizados para a seleção deles foram: terem participado do *Cell Tracking Challenge* durante as edições do ISBI de 2013 a 2015; terem ficado entre os três primeiros colocados; e terem trabalhado com pelo menos um dos sete *datasets* utilizados neste trabalho. Apesar do *Cell Tracking Challenge* ter continuado ativo até o momento de execução deste trabalho, foram escolhidos apenas os resultados destes três anos devido ao fato deles estarem presentes em uma publicação em uma revista reconhecida, enquanto que os resultados mais recentes estão disponíveis apenas no site do *Cell Tracking Challenge* e são alterados constantemente, o que ocasionaria constantes mudanças na escrita desta seção.

Na Tabela 2 é feito um comparativo dos métodos utilizados pelos competidores do *Cell Tracking Challenge* e no restante do capítulo os métodos utilizados por cada um deles são descritos com mais detalhes. As informações descritas neste capítulo foram obtidas por meio de [Ulman et al. \(2017\)](#).

3.1 KTH-SE

Esse grupo inicia com um pré-processamento no qual todas as imagens são convertidas para imagens de 64 bits com intensidade de saturação de 1.

Para a fase de segmentação, são utilizados três algoritmos diferentes para segmentar as imagens e que depois passam por um pós-processamento para extrair as regiões das células. Os parâmetros escolhidos para o algoritmo de segmentação são otimizados de forma manual ou, quando possível, utilizando um algoritmo de busca. Mas, em alguns casos, os melhores resultados foram obtidos com a otimização manual.

O primeiro método de segmentação é o *bandpass segmentation*, no qual é utilizado o algoritmo apresentado em [Maška et al. \(2014\)](#). Esse método é utilizado para todos os *datasets*

Tabela 2 – Comparação Entre os Métodos Utilizados Pelos Competidores do *Cell Tracking Challenge*

Competidores	Quantidade de métodos de rastreamento implementados	Abordagem de rastreamento utilizada	Datasets Suportados	Suporte a Divisão Celular
KTH-SE	2	<i>tracking by detection</i>	2D/3D	Sim
LEID-NL	1	<i>tracking by evolution</i>	2D/3D	Sim
IMCB-SG	2	<i>tracking by detection</i>	2D/3D	Sim
HD-Har-GE	1	<i>tracking by detection</i>	2D/3D	Sim
CUNI-CZ	1	<i>tracking by detection</i>	2D/3D	Sim
MU-CZ	1	<i>tracking by evolution</i>	2D/3D	Sim
FR-Ro-GE	1	<i>tracking by detection</i>	2D/3D ¹	Não
UZH-CH	1	<i>tracking by detection</i>	2D/3D ¹	Sim
NOTT-UK	1	<i>tracking by detection</i>	2D/3D	Sim
KIT-GE	1	<i>tracking by detection</i>	2D/3D	Não
FR-Be-GE	1	não identificado	2D/3D ¹	Sim

¹Não foi possível confirmar.

do tipo Fluo e o PhC-C2DL-PSC. A filtragem é executada fazendo uma convolução da imagem original com dois filtros gaussianos diferentes e em seguida executando um *threshold*. Para evitar a baixa segmentação de objetos muito claros, é utilizado um corte de intensidade, no qual todos os valores de pixel acima de um valor de intensidade I_{\max} recebem o valor desse I_{\max} . Há *datasets* em que há regiões do *background* muito claras na parte superior da imagem e que não são removidas totalmente pelo *bandpass filter*. Para resolver isto, é calculado um *background* da imagem pegando o valor de intensidade mínimo da posição de cada pixel através da dimensão do tempo da sequência de imagens. Em alguns *datasets* as propriedades dos ruídos são diferentes nas diversas dimensões da imagem, então é utilizado um filtro de mediana com dimensões 5x1x1 para reduzir o ruído antes de aplicar o *bandpass filter*. Em alguns *datasets* também são utilizados valores diferentes de desvio padrão para a primeira e última imagem da sequência, e é utilizada uma interpolação linear para calcular valores diferentes para cada imagem entre elas.

O segundo método de segmentação é o *variance segmentation*. Ele consiste em calcular uma textura da imagem representando a variação de intensidade na região ao redor do pixel na imagem original. Além disso, é utilizado um peso ao redor dos pixels utilizando um *kernel* gaussiano. Em seguida, é aplicado um *threshold*. Para se livrar das características do *background*, é feito primeiro a subtração do *background*. A imagem do *background* é calculada como sendo a mediana da intensidade para cada posição de pixel, levando em consideração a dimensão do tempo da sequência.

O terceiro método de segmentação é o *ridge segmentation*. Ele é inspirado no algoritmo usado para segmentar fibras musculares em [Sertel et al. \(2011\)](#). Primeiro é aplicado um *ridge detection filter* similar ao filtro descrito em [Sertel et al. \(2011\)](#), para realçar os limites entre as células. A detecção do *ridge* é feita pela suavização da imagem com *kernels* gaussianos com desvios padrões de 5, 6, 7, 8, 9, e 10 pixels e executando o Hessian ([DUDA; HART;](#)

STORK, 2000) em cada pixel das seis imagens resultantes. Depois de obter a imagem *ridge*, as intensidades são transformadas utilizando uma função, e depois divididas pela intensidade média da imagem transformada. Em seguida, é realizado um *threshold*, e a imagem é esqueletizada para extrair os limites das células. Para determinar quais regiões resultantes são células e quais são *background*, é calculada a variação local da imagem na qual cada valor de pixel representa a variação da amostra em uma vizinhança 9x9 do pixel correspondente na imagem original. Regiões com uma variação média local acima de 0.0005 são consideradas como células. Para preencher lacunas nos limites esqueletizados, todos os pontos finais do esqueleto são detectados e os pares são conectados por linhas retas. Os pontos finais são conectados se eles não estiverem separados por mais de 50 pixels, e se a linha adicionada cortar um único segmento, sem gerar um fragmento menor que 7500 pixels. Se uma das novas regiões se tornar uma região do *background*, cujo limite de tamanho é definido como 200 pixels, então a operação não divide uma célula em duas. Após juntar todos os pontos finais, as rachaduras das regiões são removidas pela remoção de todos os pixels do limite, que estão na fronteira de uma única região. Em seguida, todas as regiões de *background* e os pixels das bordas são mesclados para gerar uma única região de *background*. Finalmente, as regiões de células com menos de 7500 pixels adjacentes são mescladas até que todas as regiões de células tenham pelo menos 7500 pixels ou sejam cercadas por pixels do *background*.

No pós-processamento da segmentação, para quebrar as regiões com múltiplas células em regiões de células individuais, é utilizado um algoritmo de *watershed*. Os valores de pixel na transformada da distância são as distâncias euclidianas para os pixels mais próximos do *background*. Nas imagens 3D, para o eixo z, em que a altura do voxel é diferente da largura, é utilizada a *anisotropic distance transform* (MISHCHENKO, 2012). Essa abordagem leva em consideração que as distâncias entre os voxels vizinhos entre dois planos z distintos são diferentes quando comparados com as distâncias entre os voxels vizinhos no mesmo plano z. Nos *datasets*, em que as distâncias entre os planos z são muito grandes, são inseridos planos virtuais no eixo z para poder calcular a distância.

Na fase do rastreamento, para todos os *datasets*, exceto o Fluo-N3DL-DRO, é aplicado o *global track linking algorithm* (MAGNUSSON et al., 2015) diretamente nas regiões de células detectadas. Para o Fluo-N3DL-DRO é utilizado um novo algoritmo de pré-processamento de detecção (MAGNUSSON; JALDEN, 2015), que leva em consideração a natureza dinâmica do movimento do núcleo da célula pré-processando os locais detectados usando um *Gaussian Mixture Probability Hypothesis Density (GM-PHD) filter* (VO; MA, 2006). Tendo pré-processado as localizações, as associações das células são feitas usando o *track linking algorithm* apresentado em Magnusson et al. (2015).

Como pós-processamento do rastreamento, para remover erros de segmentação devido à segmentação excessiva causada pelo *watershed*, os fragmentos das regiões sem células são mescladas iterativamente em regiões adjacentes com células depois que o rastreamento é concluído.

3.2 LEID-NL

Desenvolveu um método baseado na abordagem de *tracking by evolution*, no qual cada objeto que está sendo rastreado é representado por um modelo, que é ajustado aos dados a cada momento. O modelo utilizado foi o de *level-sets* devido à sua capacidade de fornecer uma segmentação de alta qualidade para uma ampla variedade de dados de microscopia e mudança natural da topologia das células.

Inicialmente é feito um pré-processamento de todas as imagens utilizando o método *PURE denoising* por meio do *plugin* correspondente do ImageJ (SCHNEIDER; RASBAND; ELICEIRI, 2012). Em seguida, a segmentação inicial, assim como a detecção da entrada das células, é feita baseada numa estimativa grosseira do *background* da imagem. Uma grande variedade de métodos de segmentação binária é utilizada, em particular vários *thresholds*. O método de segmentação mais apropriado para cada *dataset* foi selecionado de modo empírico. A separação de duas células que estão coladas uma na outra é feita baseada no valor mínimo da transformada de Radon (BARRETT, 1988) da intensidade da imagem na janela local que contém as duas células.

Para o rastreamento, o modelo no qual o algoritmo é baseado assume que há sobreposição suficiente entre duas regiões ocupadas por uma célula a cada dois instantes do vídeo. Porém, em alguns casos, essa condição é violada, sendo necessário estimar a nova posição de cada célula no instante seguinte baseada na posição do instante atual. Essa operação é feita antes da evolução de cada contorno, e consiste de dois passos principais: detecção do marcador; e estimativa conjunta da nova posição de todos os objetos.

Na divisão celular, duas células filhas podem não se distanciarem uma da outra, para lidar com esses casos, um pós-processamento é realizado, no qual um novo contorno é criado utilizando menos restrições a partir do estado atual. Se os contornos se dividirem em duas partes separadas com a adição desse contorno adicional então eles são marcados como pertencentes a objetos divididos, caso contrário, o contorno antigo é mantido.

3.3 IMCB-SG

Utiliza duas abordagens distintas: uma para os *datasets* 2D, em que é realizado o rastreamento seguido da segmentação; e outra para os *datasets* 3D, em que o processo é inverso, primeiro é realizada a segmentação e em seguida o rastreamento.

3.3.1 Datasets 2D

Para o rastreamento, primeiro é utilizado um filtro gaussiano para suavizar as imagens no domínio espacial. Isso é feito para remover os ruídos e realçar a região dos objetos de interesse. O tamanho da janela e o desvio padrão do filtro gaussiano são definidos baseado na estrutura

das células ou do núcleo nas imagens. Em seguida, operações morfológicas são utilizadas para remover o ruído do *background*. Para imagens de baixo contraste eles utilizam a equalização do histograma e técnicas de ajuste de contraste para realçar as imagens.

O segundo passo no rastreamento é alinhar as imagens 2D em pilhas 3D cuja a terceira dimensão representa o tempo. Após alinhar as imagens, é usado uma convolução baseada no filtro gaussiano para suavizar as imagens ao longo do tempo. Isso é feito para recuperar objetos faltantes. Após esses dois passos, é usada uma *extended maxima transform* para localizar sementes baseada no gradiente da imagem.

No último passo do rastreamento, os vestígios das sementes 2D mencionadas formam segmentos 3D que são as trajetórias do rastreamento procuradas. Em seguida, é aplicada uma operação de abertura para remover pequenos segmentos 3D que são considerados ruídos ou artefatos causados pelo processamento. O rastreamento é realizado pelo rastreamento das trajetórias dos segmentos 3D remanescentes. O movimento celular e os eventos de mitose são todos incorporados nas trajetórias.

Para a fase de segmentação, é utilizado o *threshold* de Otsu (OTSU, 1979). Em seguida, é utilizada uma abertura para remover falsos objetos. Com as imagens binárias obtidas, as sementes em cada imagem do tempo e o gradiente da imagem após a suavização gaussiana nos domínios espacial e temporal são combinados para usar o algoritmo de *watershed* (GONZALEZ; WOODS, 2002) para obter a segmentação final. O algoritmo de *watershed* é usado para resolver o problema das células que se grudam umas nas outras.

3.3.2 Datasets 3D

A realização da segmentação é feita de forma diferente para os *datasets* normais e para os *datasets* com baixa relação sinal-ruído com intensidade altamente não-uniforme.

Para os *datasets* normais, é utilizado um *kernel* gaussiano 3D para suavizar as imagens. Os parâmetros do *kernel* gaussiano são definidos de acordo com o tamanho da célula ou do núcleo da célula. Em seguida, é gerado um histograma para gerar o valor do limiar do *threshold* a ser aplicado. Logo após, é aplicada uma abertura para remover pequenos objetos. Além disso, é utilizada uma *distance transform* para calcular a distancia euclidiana. A distância é utilizada como entrada para a *extended maxima transform* para encontrar as sementes 3D que serão utilizadas no algoritmo de *watershed* para finalizar a segmentação.

Para os *datastes* com baixa relação sinal-ruído com intensidade altamente não-uniforme, primeiro é realizado uma interpolação linear para gerar pilhas de imagens com amostragem reduzida. Em seguida, é realizada uma suavização gaussiana no domínio espacial 3D. Ao mesmo tempo, são utilizadas operações morfológicas para remover o *background* e ajustes de contraste para realçar as imagens. Depois disso, as imagens com amostragem reduzida são interpoladas para voltarem para o tamanho original. Logo após, é realizado um *threshold* para

obter as imagens binárias, e é usada uma função de distância ou de intensidade de imagem para detectar as sementes para utilizar a *extended maxima transform*. A segmentação final é realizada utilizando um *watershed* com as sementes encontradas.

Para a fase de rastreamento, são utilizadas duas abordagens. A primeira é um algoritmo baseado na abordagem do vizinho mais próximo. Primeiro é definida uma região correspondente à célula na primeira imagem de tempo. De acordo com a quantidade de possíveis objetos na região na imagem de tempo seguinte, é definido se é a mesma célula ou mitose. Uma técnica recursiva de rastreamento é usada para construir a árvore de linhagem de mitoses da célula.

A segunda abordagem é baseada na correspondência global usando o algoritmo de Hungarian (KUHN, 1955), que combina a distância euclidiana e a sobreposição do volume. Então, os seguintes mapeamentos são seguidos:

- a) Um-para-um: é considerada a mesma célula.
- b) Muitos-para-um: é escolhida a que tiver o menor custo e o restante é considerada como se tivesse saído nesse instante do vídeo.
- c) Um-para-muitos: é considerado um evento de mitose e se a possível correspondência for maior que dois então a situação de uma nova célula surgindo é considerada.

3.4 HD-Har-GE

Realiza um pré-processamento utilizando um filtro gaussiano ou um filtro de mediana para reduzir os ruídos da imagem. Caso haja manchas brancas dentro dos núcleos da célula ou contrastes fortemente variáveis de objetos vizinhos, um corte de intensidade é realizado usando o *threshold* de Otsu ou o *threshold* de entropia de Renyi (SAHOO; WILKINS; YEAGER, 1997).

Após o pré-processamento, um *threshold* com região adaptativa é aplicado para obter uma segmentação inicial. A abordagem inicia com regiões pequenas da imagem e computa *thresholds* locais dentro de regiões com um tamanho similar ao tamanho médio do diâmetro do objeto. A sensibilidade da abordagem pode ser adaptada aos dados da imagem baseado na variância mínima da intensidade dentro das regiões da imagem. Se a variância dentro de uma respectiva região exceder o valor mínimo então um *threshold* local é utilizado, caso contrário, um *threshold* global é usado. O resultado da segmentação é reforçado pelo filtro de mediana e pelo preenchimento dos buracos. Para separar células coladas, é utilizado o algoritmo de *watershed* após ter sido realizada uma transformada da distância euclidiana.

A abordagem do rastreamento explora a informação da segmentação das células e consiste de três passos principais:

1. Determinação das correspondências um-para-um: é utilizado um procedimento de otimização local, onde para cada objeto, hipóteses, chamadas de *triplets*, são geradas com todos os antecessores e sucessores em potencial dentro de uma distância euclidiana pré-fixada. Estes *triplets* são classificados e comparados com outros *triplets* possíveis. A classificação das hipóteses é determinada baseada numa função de custo, que inclui a distância euclidiana, a semelhança morfológica, assim como a suavidade da trajetória. Os pesos para esses três componentes são especificados baseados nas propriedades do tipo da imagem.
2. Detecção das mitoses e estabelecimento das correspondências um-para-muito: O aparecimento de objetos são investigados para detectar os eventos de mitose. Dependendo do tipo de coloração é explorado diferentes medidas para a detecção da mitose:
 - a) A razão distância-sobreposição: determina a razão entre a área de sobreposição de uma célula filha e da célula mãe em potencial e a distância entre seus centroides.
 - b) Uma medida de probabilidade baseada na morfologia do objeto: leva em conta os tamanhos e as intensidades médias da célula filha e da célula mãe em potencial normalizadas pelos valores médios de toda a população.
 - c) Uma combinação das duas medidas: explora ambas as situações descritas anteriormente.

Se um evento de mitose é detectado, as trajetórias da nova célula filha e da célula mãe são fundidas.

3. Detecção e junção das trajetórias que não cobrem todos os *frames* de uma sequência de imagens: todas estas trajetórias são consideradas e fundidas se estiverem em uma vizinhança espaço-temporal próxima.

3.5 CUNI-CZ

Inicialmente realiza um pré-processamento utilizando um *blur* com um *kernel* gaussiano para diminuir o ruído e suavizar os limites dos objetos segmentados. Em seguida, o processo de segmentação é iniciado criando um mapa discreto de *thresholds*, cujo mapa é criado de modo que uma janela de tamanho fixo percorre toda a imagem, e cada espaço percorrido possui seu próprio valor de *threshold*. Os valores de *threshold* são selecionados através de um método de seleção iterativo baseado no algoritmo de *k-means*. Em seguida, os objetos com buracos são preenchidos utilizando uma *fill hole transform*.

Para resolver o problema de células coladas umas nas outras é utilizada uma *watershed transform*, que é aplicado após uma das seguintes situações: uma abertura por reconstrução da imagem da câmera após a aplicação do *blur*; ou uma abertura por reconstrução da *distance transform* após a aplicação do *threshold*. Por fim, apenas os objetos maiores que um determinado tamanho de área são levados em consideração.

Para a fase de rastreamento é utilizada a abordagem do vizinho mais próximo, onde o ponto de referência utilizado é o centro de gravidade do objeto. As distâncias são medidas par a par entre as posições dos objetos na imagem atual e as posições dos objetos nas imagens anteriores, e então, transformadas em probabilidades, cujos valores mais altos correspondem às melhores correspondências de posição. Combinando as probabilidades com os pesos e os limiares definidos pelo usuário, tem-se as seguintes situações:

- a) Muitos-para-um (múltiplas células se sobrepondo): o objeto na imagem atual é marcado com o rótulo do objeto mais próximo da imagem anterior. Para os outros objetos da vizinhança é seguido o caso um-para-nenhum.
- b) Um-para-um (objeto migra para uma nova posição): o *track* do objeto é mantido e sua posição é atualizada.
- c) Um-para-nenhum (objeto desaparece do campo de visão): a posição não é atualizada, mas o peso do objeto é diminuído em 5%.
- d) Nenhum-para-um (objeto entra no campo de visão): um novo objeto é criado.
- e) Um-para-muitos (objeto se divide em dois ou mais objetos): dois ou mais objetos são criados e é atribuído o peso 0 para o objeto original.

3.6 MU-CZ

Desenvolveu um método baseado na abordagem de *tracking by evolution*. O método utilizado é baseado na minimização do modelo de Chan-Vese em um *framework* de configuração de nível rápido que é integrado com uma prévia topológica, permitindo o rastreamento simultâneo de múltiplas células ao longo do tempo. Inicialmente, para diminuir a quantidade de ruído, todas as imagens são pré-processadas usando um filtro gaussiano com um desvio padrão de 1 pixel em cada direção. Foi empregada uma rotina de separação de *clusters* baseada na evolução de contornos ativos implícitos topologicamente inflexíveis.

A abordagem utilizada assume que as novas células entrando no campo de visão são componentes de fronteira disjuntos com as células existentes. Porém, não necessariamente, as novas células que estão entrando tocam a borda da imagem. Então essa abordagem foi modificada omitindo a necessidade destas células estarem na borda em uma máscara obtida usando um *weighted 2-means clustering*. Para compensar o fenômeno da variação da intensidade da fluorescência com o passar do tempo dentro dos núcleos individuais, o que acontece especialmente pouco antes e depois do evento de mitose, o peso do *background* no modelo Chan-Vese é temporariamente multiplicado por uma constante definida pelo usuário. Como pós-processamento é realizada a eliminação das células que estão fora do FoI, ou seja, estão fora da área da imagem a ser analisada.

3.7 FR-Ro-GE

Realizou um pré-processamento, no qual os valores em escala de cinza de todos os *datasets*, exceto o *dataset* Phc-C2DH-U373, foram normalizados para uma mediana na faixa de unidade e zero para compensar as mudanças na iluminação durante a gravação. Além disso, dois dos *datasets* (DIC-C2DH-HeLa e Fluo-C2DL-MSD) foram *downscaled* para alcançar um campo de visão abrangente e compensar as diferentes resoluções da gravação.

Para realizar a segmentação foi utilizada uma *u-shaped deep convolutional network* cuja arquitetura é uma extensão da *fully convolutional network*.

Para o rastreamento é utilizado um algoritmo *greedy*. Cada segmento no *frame* t propaga seu rótulo para o segmento no *frame* $t+1$ com a sobreposição mais alta (medida como a interseção sobre a união). Se o segmento no *frame* $t+1$ recebe múltiplos rótulos, o segmento no *frame* t com a sobreposição mais alta é utilizada e os outros rótulos são descartados. Se o segmento não recebe nenhum rótulo, um novo rótulo é associado. Adicionalmente, os segmentos que estejam fora do FoI são descartados, mas suas informações continuam sendo utilizadas para que a associação de pai seja adicionada nos casos em que os segmentos, que foram rastreados no campo de visão como um todo, entrem novamente no FoI.

3.8 UZH-CH

Primeiro realiza um pré-processamento para aumentar a relação sinal-ruído, no qual o contraste é ajustado mapeando os valores de intensidade da imagem de entrada de modo que 1% dos dados sejam saturados em baixos e altos valores de intensidade. Em seguida, é aplicado um *blur* com um filtro gaussiano. Por fim, é calculada a intensidade média e subtraída da imagem, onde os valores negativos de intensidade são substituídos pelo seu valor absoluto.

O processo de segmentação é feito utilizando *watershed*, onde as sementes são calculadas utilizando o complemento da imagem e calculando a transformada da distância negativa. Em seguida é aplicado um *threshold* de Otsu com decaimento. E para eliminar pequenos artefatos, objetos menores que 10 pixels são apagados.

O rastreamento é feito calculando a distância entre os objetos de um instante (t) e o instante seguinte ($t+1$), no qual um valor máximo de distância (50 pixels) é aplicado para cada objeto. A quantidade máxima de possíveis vizinhos é limitado a 5 nos *datasets* 2D. Dependendo do número de possíveis vizinhos e seus tamanhos relativos comparados ao objeto, são definidos três cenários: desaparecimento; é a própria célula; e divisão celular. Quando nenhum vizinho é detectado ou apenas um vizinho é detectado, mas possui um tamanho maior que duas vezes do objeto, é considerado um desaparecimento. Quando um único vizinho é detectado e possui um tamanho aceitável (até duas vezes menor que o objeto), é considerado como a própria célula. Sempre que possuir mais de dois possíveis vizinhos são criadas todas as possíveis combinações

de pares entre os vizinhos e então é checada a média do centroide do par mais próximo do objeto. Se o tamanho total do par mais próximo não for maior que duas vezes do tamanho total do objeto então é aceito como divisão celular.

3.9 NOTT-UK

Inicia com um pré-processamento utilizando um método existente de realce baseado em operações morfológicas, *Top-Hat* e *Bottom-Hat*.

O algoritmo de segmentação foi desenvolvido usando técnicas de *threshold* local. Para realçar as células das imagens, primeiro a imagem é separada em sub-imagens. Então, o método detecta as células, ou parte delas, em cada sub-imagem comparando os desvios padrões com o desvio padrão da imagem completa. Se houver objetos detectados no processamento de sub-imagem, o método executará a mesma abordagem de realce usada na etapa de pré-processamento para melhorar profundamente a qualidade da imagem. Em seguida, o *threshold* de Otsu é usado para segmentar a sub-imagem. Depois que todas as sub-imagens forem processadas, os resultados delas são colocados juntos para gerar o resultado final. Para alguns *datasets*, mesmo com os realces, ainda há células em fragmentação. Nestes casos, o algoritmo detecta os fragmentos das células calculando sua redondeza. Se o parâmetro é mais baixo que um valor pré-definido, a área detectada será realçada e segmentada para regenerar a forma da célula que está sendo processada. Diferente dos outros *datasets*, a abordagem de segmentação para o *dataset* Fluo-C2DL-MSD é feito usando um *threshold* global. O pré-processamento é o mesmo para este *dataset*. E em seguida, é utilizado o *threshold* de Otsu para cada *frame* para gerar os resultados da segmentação.

O rastreamento é feito por meio da associação *frame a frame*. Uma sequência segmentada dos instantes de tempo, cujas células são detectadas em cada *frame*, é a entrada para o algoritmo. O centro de gravidade da célula é escolhido para representar a posição da célula. Consequentemente, o método é desenvolvido para calcular primeiro os centros das células no *frame* que está sendo processado e nos *frames* anteriores. Então, para cada célula no *frame* atual, a abordagem calcula a distância euclidiana do centro da célula para as células no *frame* anterior, e associa à célula mais próxima dela. Após a célula ser associada aos *frames* vizinhos, se duas ou mais células estão apontando para a mesma célula no *frame* anterior, é considerada a ocorrência de uma divisão celular, e novos rótulos são associados para cada célula no processamento da imagem (um-para-muitos). Em adição, se a distância mínima excede um limiar pré-definido, que é o raio da célula, a célula que está sendo processada é considerada como a aparição de uma nova célula no *frame* atual e é marcada com um novo rótulo (nenhum-para-um). De modo recíproco, as células no *frame* anterior são marcadas como desaparecidas, se não existe nenhuma célula que corresponda a ela no *frame* atual (um-para-nenhum). Além do rastreamento *frame a frame*, para os *datasets* 3D, como o Fluo-N3DH-CHO, as correspondências entre camadas diferentes também são necessários. Para conseguir isso, o método começa com a segmentação das

diferentes camadas. Em seguida, resume os resultados em uma imagem e executa o rastreamento de células com o *frame* vizinho. De acordo com o grafo rotulado obtido, o método finalmente marca as células nas diferentes camadas com os rótulos correspondentes.

3.10 KIT-GE

Inicia com um pré-processamento utilizando um filtro de mediana 2D para reduzir os ruídos da imagem, e um filtro gaussiano para suavizar o resultado da segmentação. Para as imagens 3D, o filtro de mediana foi executado individualmente em cada *frame* da imagem. Para o *dataset* Fluo-N2DH-GOWT1 uma operação de fechamento adicional foi usada para evitar buracos na segmentação. Uma versão das imagens de entrada com amostragem reduzida foi utilizada para detectar as sementes. As sementes foram detectadas pela identificação do máximo local na 8-vizinhança e 26-vizinhança de cada pixel para os *datasets* 2D e 3D, respectivamente, dentro de uma *Laplacian-of-Gaussian (LoG) space-scale maximum projection*, que é calculada iterativamente usando imagens filtradas com LoG de diferentes escalas discretas. Eles usaram uma detecção máxima não estrita para evitar erros de detecção causados por platôs de intensidade e pontos de semente redundantes mesclados com base em um critério de distância mínima esperada. Em cada local identificado, a intensidade média de uma janela 7x7 foi calculada e usada para descartar pontos de semente de baixa intensidade com um limite binário otimizado semi-automaticamente.

Em seguida, a segmentação é executada em cada imagem individualmente usando o algoritmo de segmentação de TWANG. Em seguida, uma nova imagem é formada baseada no produto escalar ponderado da gaussiana com o vetor gradiente da intensidade normalizado de cada pixel. Nesta imagem transformada, as regiões de transição entre núcleos individuais obtiveram valores de baixa intensidade, enquanto os pixels pertencentes ao núcleo atualmente considerado obtiveram valores de alta intensidade que puderam ser facilmente separados do fundo usando um limiar binário adaptativo (método de Otsu).

Para o rastreamento, a identificação das associações temporais dos núcleos detectados foi derivada com a *toolbox* de rastreamento contida na *toolbox* de código aberto Gait-CAD do MATLAB. Essencialmente, os centroides dos segmentos identificados foram rastreados pela identificação dos vizinhos mais próximos nos quadros subsequentes. As correspondências só foram consideradas válidas se a distância máxima não fosse excedida. Se a relação de distância entre o vizinho mais próximo e o segundo vizinho mais próximo fosse suficientemente pequena, o núcleo era considerado um potencial candidato à divisão celular. As rotinas de pós-processamento para detecção de divisão de células e a fusão de *tracks* fragmentados, no entanto, ainda não tinham sido desenvolvidos.

3.11 FR-Be-GE

Primeiro é realizado um pré-processamento normalizando as intensidades da imagem para o intervalo $[0, 1]$. Então, o *background* das imagens são corrigidos pela subtração da imagem original pela imagem suavizada. A suavização da imagem é realizada utilizando um *kernel* gaussiano com desvio padrão de 20 pixels.

A abordagem utilizada para a segmentação explora o fato de que as bordas das células verdadeiras em microscopia de fase de contraste positivo sempre aparecem como uma transição do escuro para o claro na direção de dentro para fora. Isso significa que todas as bordas com a transição inversa (do claro para o escuro) definitivamente não são as bordas das células procuradas. Eles definiram uma função de segmentação de energia para uma máscara e uma dada imagem. A função contém um custo de dados que depende da intensidade, e um custo de aresta que depende do gradiente da intensidade da borda da máscara na direção de dentro para fora. Por fim, eles, opcionalmente, aplicam um algoritmo de preenchimento de buracos e descartam segmentos pequenos que sejam menores que um valor de área definido.

O rastreamento consiste em duas partes: a propagação da segmentação, que promove uma segmentação temporariamente consistente por propagar a informação da segmentação para os *frames* subsequentes; e a propagação dos rótulos, que transfere os rótulos de cada segmento para os *frames* subsequentes usando uma *greedy association*.

Na propagação da segmentação, uma segmentação *min-cut* produz a máscara de segmentação binária. A informação da segmentação é propagada do *frame* t para o *frame* $t+1$ de duas maneiras:

- a) *Foreground propagation* (FP): A máscara erodida é definida como uma forte limitação para a segmentação *min-cut* no próximo *frame*. Isto adiciona robustez para o termo da região no caso de evidência insuficiente do *foreground*. O tamanho escolhido para a erosão deve ser no mínimo do mesmo tamanho do movimento esperado da borda dos pixels do objeto entre os *frames*. A erosão é executada usando um elemento estruturante em forma de disco.
- b) *Non-mergin constraint* (MF): Se for possível assumir que as células não se misturam, é razoável evitar que objetos separados sejam misturados no próximo *frame*. Isso é feito executando uma transformação de distância na máscara de segmentação e aplicando um *watershed* nos locais dos objetos. Os limites das “regiões de suporte” computadas de cada objeto são definidas como restrições de *background*.

Na propagação dos rótulos, um algoritmo *greedy* é utilizado para propagar os rótulos. Cada segmento no *frame* t propaga seu rótulo para o segmento no *frame* $t+1$ com a sobreposição mais alta (medida como a interseção sobre a união). Se o segmento no *frame* $t+1$ recebe

múltiplos rótulos, o segmento no *frame* t com a sobreposição mais alta é utilizada e os outros rótulos são descartados. Se o segmento não recebe nenhum rótulo, um novo rótulo é associado. Adicionalmente, os segmentos que estejam fora do FoI são descartados, mas suas informações continuam sendo utilizadas para que a associação de pai seja adicionada nos casos em que os segmentos, que foram rastreados no campo de visão como um todo, entrem novamente no FoI.

4

Resultados

Neste capítulo serão apresentadas cada uma das fases necessárias para a realização do rastreamento celular com suas respectivas propostas de resolução, assim como a discussão dos resultados obtidos durante o passo de validação das fases de segmentação e rastreamento.

Para a realização deste trabalho foram necessárias três fases principais:

- a) Segmentação: Fase responsável por segmentar as células.
- b) Rotulação: Fase responsável por atribuir um rótulo que identifique cada célula de forma única.
- c) Rastreamento: Fase responsável por estabelecer as correspondências entre as células identificadas nas imagens.

Para o desenvolvimento de cada uma das fases foi utilizada a VisionGL ([DANTAS; LEAL; SOUSA, 2016](#)), que é uma biblioteca voltada para o processamento de imagens multidimensionais.

4.1 Primeira Fase: Segmentação

Devido à diversidade dos *datasets* utilizados, cada um com suas peculiaridades, mais de 60 combinações de operações de segmentação foram utilizadas até que fossem encontradas as combinações mais adequadas para cada *dataset* em específico.

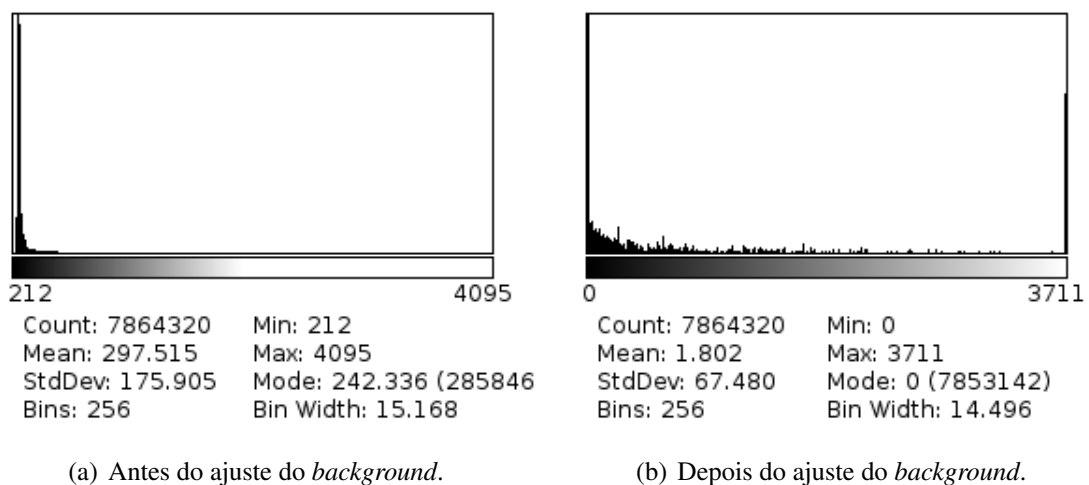
As combinações de operações utilizadas em cada *dataset*, assim como seus respectivos parâmetros, foram escolhidos na base de tentativa e erro até que fosse encontrada a combinação que melhor satisfizesse o objetivo inicial. O objetivo inicial era obter o maior número de células possíveis que não estivessem grudadas umas nas outras no instante zero do vídeo. Isso

era necessário para que fosse possível diferenciar nos instantes seguintes do vídeo quando ocorressem casos de colisão entre células e casos de mitose.

Para testar as combinações de operações, foram selecionadas as imagens que representavam os instantes zeros dos vídeos 01 de cada *dataset*. Após verificar que o objetivo inicial tinha sido atingido, o método era aplicado para o restante das imagens do vídeo 01 e aplicado a todas as imagens dos outros vídeos.

Os testes iniciais foram realizados utilizando o *dataset* Fluo-C3DL-MDA231, cuja primeira combinação de operações consistia apenas de operações de reconstrução por abertura e de reconstrução por fechamento. A partir dessa primeira combinação, foi possível perceber que as regiões pretas da imagem eram constituídas de pixels com valores de intensidade diferentes de zero, como pode ser visto através dos histogramas (ver Figura 5), o que dificultava o processo de segmentação (ver Figura 6). Dessa forma, foi identificada a necessidade de realizar um ajuste do *background* para que estes pixels fossem eliminados. Para realizar esses ajustes foram testadas abordagens com aplicações de *thresholds*, e abordagens com operações de erosão, dilatação e *top-hat*, porém, nenhuma dessas abordagens atingiu o objetivo inicial para esse *dataset*. Após várias tentativas, chegou-se ao método que melhor realizava este ajuste. O método consiste em calcular o valor da moda e utilizá-lo para realizar uma subtração da imagem original. A subtração era feita de modo que os valores de intensidade menores ou iguais ao valor da moda recebessem o valor 0; e os outros pixels recebessem seus próprios valores subtraindo-se o valor da moda. Após ter ajustado o *background* da imagem, deu-se continuidade aos testes até encontrar a melhor combinação de operações de segmentação para este *dataset*.

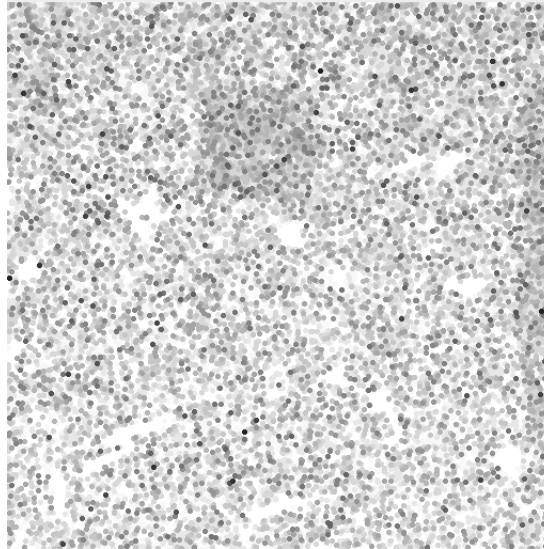
Figura 5 – Histograma da imagem “t000” do *dataset* Fluo-C3DL-MDA231



Tendo encontrado a melhor combinação para o *dataset* Fluo-C3DL-MDA231, os testes para os outros *datasets* foram iniciados utilizando essa mesma combinação, porém, os resultados obtidos ainda não atingiam o objetivo inicial para os mesmos. Dessa forma, foram testadas novas

abordagens usando esta como ponto de partida, visto que nestes *datasets* o objetivo inicial estava próximo de ser atingido.

Figura 6 – Resultado da segmentação do *frame* 9 da imagem “t000” do *dataset* Fluo-C3DL-MDA231 antes do ajuste do *background*



Os *datasets* Fluo-C3DH-H157, Fluo-N3DH-SIM, Fluo-N3DH-CHO, Fluo-C2DL-MSK e PhC-C2DH-U373 utilizaram abordagens baseadas em *threshold*, onde o valor do limiar adotado era o valor da moda da imagem, porém, esse valor não era o mais adequado. Dessa forma, o valor de limiar para cada um destes *datasets* foi gerado multiplicando o valor da moda por pesos que melhor se adequassem para os mesmos. Além disso, o valor resultante da multiplicação foi dividido pela constante 65535 para que o valor do limiar ficasse no intervalo $[0, 1]$, que é o formato aceito pela operação de *threshold* utilizada. A escolha do valor da constante deu-se pelo fato deste ser o maior valor possível que uma imagem de 16 bits pode conter.

Nas subseções a seguir serão apresentadas as combinações utilizadas para realizar a segmentação de cada *dataset*, assim como o pós-processamento utilizado em alguns deles.

4.1.1 Pós-processamento

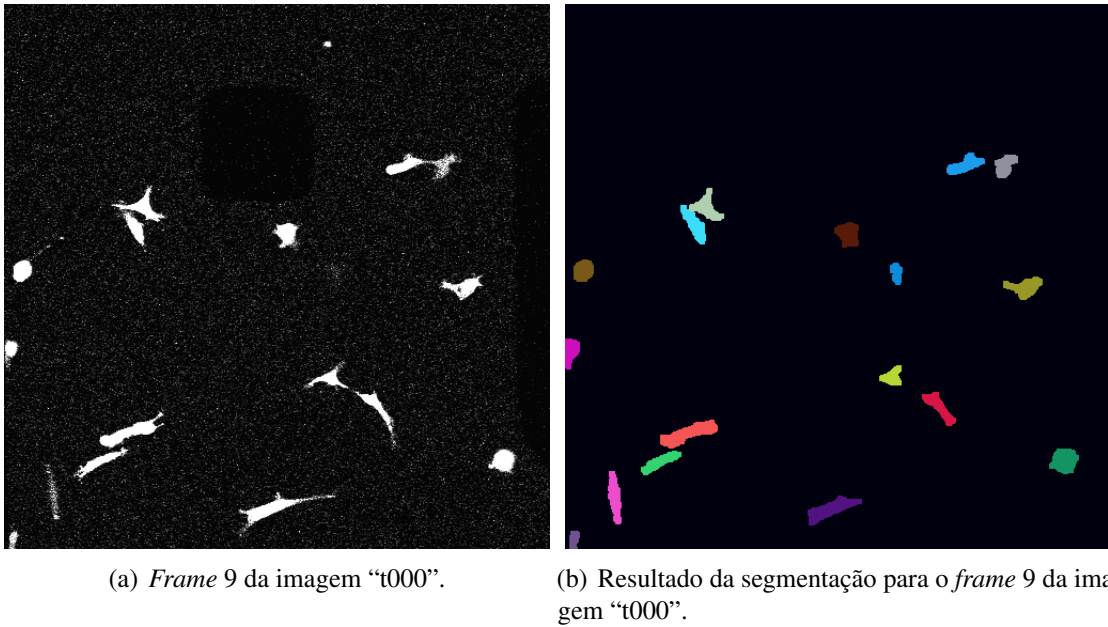
O pós-processamento consiste em eliminar resíduos que não tenham sido eliminados durante a segmentação e que não possuam o tamanho suficiente para ser considerado uma célula. Isso é feito calculando um tamanho mínimo que é baseado em 30% da média do tamanho total das células. Toda componente conexa que for menor que esses 30% será eliminada. Esse valor de 30% foi selecionado após vários testes e ter sido verificado que este valor era o suficiente para eliminar os ruídos ainda presentes na imagem.

4.1.2 *Dataset* Fluo-C3DL-MDA231

Para realizar a segmentação deste *dataset*, foi utilizada a seguinte abordagem:

1. Calcular o valor da moda da imagem e fazer com que os pixels com valores de intensidade menores ou iguais a $\text{moda} \times 1,5$, recebessem o valor 0; e os outros pixels recebessem seus próprios valores subtraindo-se o valor da moda.
2. Realizar uma erosão com um elemento estruturante (EE) em forma de cruz com dimensões 3x3 com apenas 1 iteração.
3. Realizar uma erosão com um EE em forma de quadrado com dimensões 3x3 com apenas 1 iteração.
4. Realizar uma reconstrução por abertura (RA) com um EE em forma de quadrado com dimensões 3x3 com apenas 1 iteração.

Figura 7 – Dataset Fluo-C3DL-MDA231



4.1.3 Dataset Fluo-C3DH-H157

O primeiro passo na segmentação deste *dataset* é calcular o valor da moda da imagem que será utilizado para gerar o valor do limiar utilizado pelo *threshold*, que é dado pela Fórmula 4.1.

$$\text{limiar} = \frac{\text{moda} \times 1,5}{65535} \quad (4.1)$$

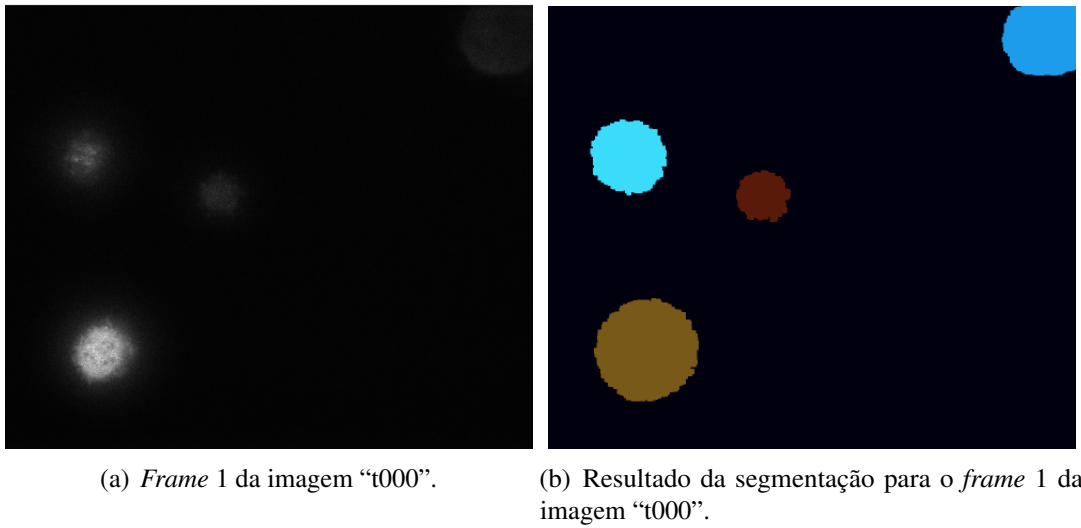
Com o resultado gerado pelo *threshold* em mãos, os seguintes operadores morfológicos são executados em sequência:

1. Uma dilatação com um EE com dimensões 3x3x3 em forma de quadrado com apenas 1 iteração.

2. Uma erosão com um EE com dimensões 3x3 em forma de quadrado com 3 iterações.
3. Um fechamento com um EE com dimensões 3x3 em forma de quadrado com 1 iteração.
4. Um fechamento com um EE com dimensões 3x3 em forma de quadrado com 2 iterações.
5. Uma abertura com um EE com dimensões 3x3 em forma de quadrado com 2 iterações.
6. Uma abertura com um EE com dimensões 3x3 em forma de quadrado com 3 iterações.

Em seguida, foi realizado o pós-processamento descrito na seção 4.1.1.

Figura 8 – *Dataset* Fluo-C3DH-H157



4.1.4 *Dataset* Fluo-N3DH-SIM

Para realizar a segmentação deste *dataset*, primeiro é realizada a seguinte sequência de operações:

1. Uma erosão com um EE com dimensões 3x3 em forma de quadrado com 25 iterações.
2. Um *top-hat* com um EE com dimensões 3x3 em forma de quadrado com 25 iterações.

Após essa sequência de operações, é realizado um *threshold* cujo valor do limiar utilizado é gerado pela Fórmula 4.2.

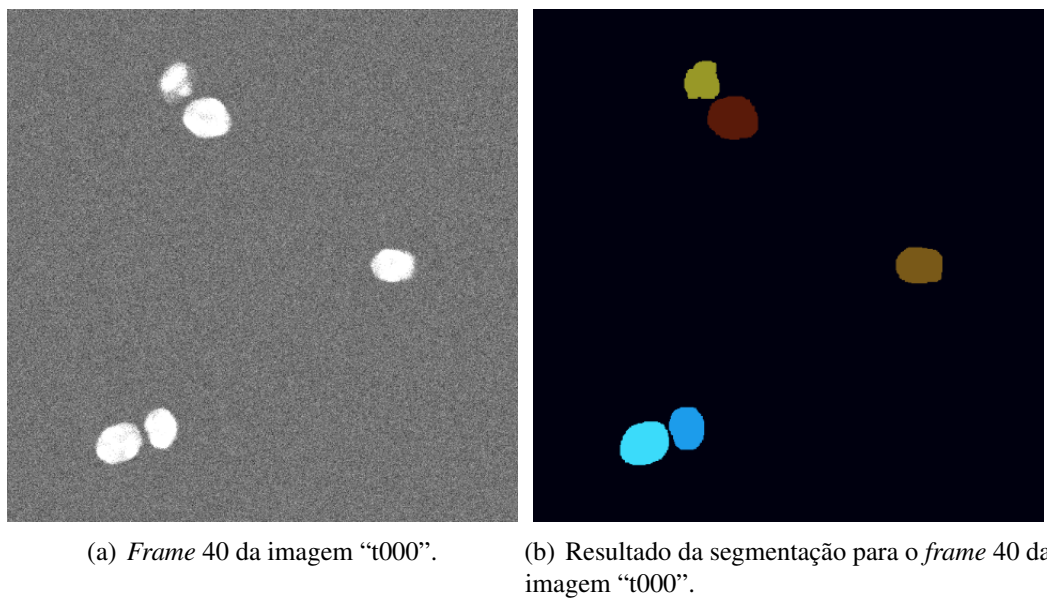
$$limiar = \frac{moda \times 2}{65535} \quad (4.2)$$

Em seguida são realizadas as seguintes operações:

1. Um fechamento com um EE com dimensões 3x3 em forma de quadrado com apenas 1 iteração.
2. Uma abertura com um EE com dimensões 3x3 em forma de quadrado com 2 iterações.
3. Uma abertura com um EE com dimensões 3x3 em forma de quadrado com 3 iterações.

Finalizando o processo de segmentação, foi realizado o pós-processamento descrito na seção 4.1.1.

Figura 9 – *Dataset Fluo-N3DH-SIM*



4.1.5 *Dataset Fluo-N3DH-CHO*

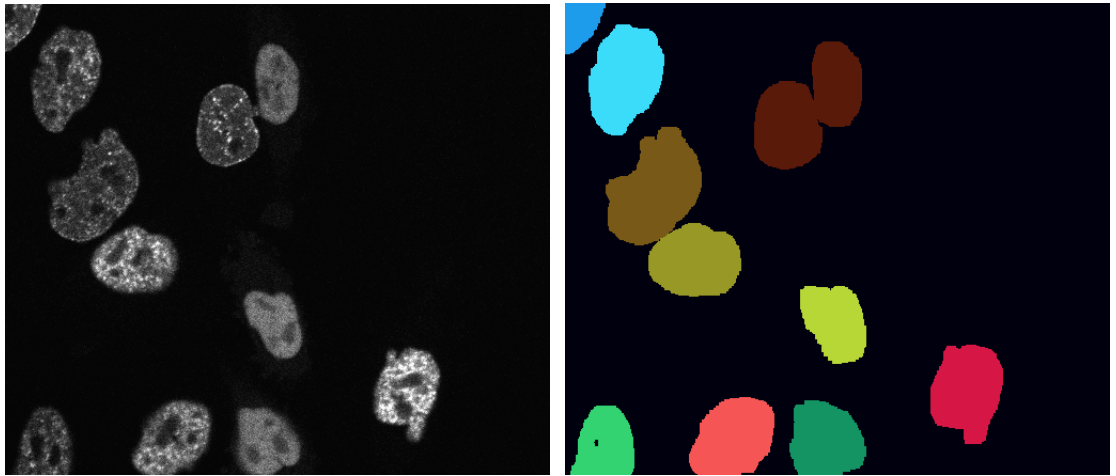
A segmentação deste *dataset* inicia-se calculando o valor da moda da imagem que será utilizado para gerar o valor do limiar utilizado pelo *threshold*, que é dado pela Fórmula 4.3.

$$limiar = \frac{moda \times 5}{65535} \quad (4.3)$$

Tendo obtido o resultado do *threshold*, são realizadas as seguintes operações:

1. Um fechamento com um EE com dimensões 3x3 em forma de quadrado com apenas 1 iteração.
2. Uma abertura com um EE com dimensões 3x3 em forma de quadrado com 2 iterações.

Para finalizar o processo de segmentação, o pós-processamento descrito na seção 4.1.1 é realizado.

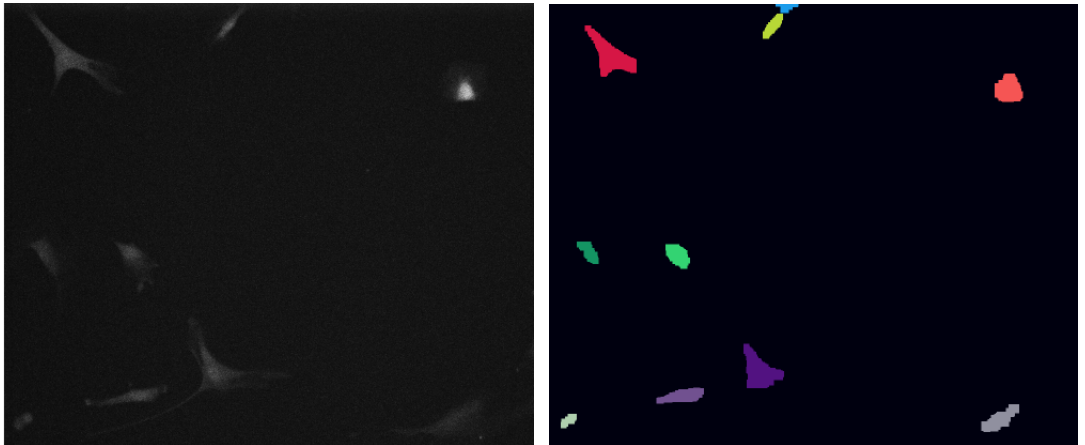
Figura 10 – *Dataset Fluo-N3DH-CHO*(a) *Frame 1* da imagem “t000”.(b) Resultado da segmentação para o *frame 1* da imagem “t000”.

4.1.6 *Dataset Fluo-C2DL-MS*

Para realizar a segmentação deste *dataset*, foi utilizada a seguinte abordagem:

1. Calcular o valor da moda da imagem e fazer com que os pixels com valores de intensidade menores ou iguais ao da moda recebessem o valor 0; e os outros pixels recebessem seus próprios valores subtraindo-se o valor da moda.
2. Realizar uma erosão com um EE em forma de quadrado com dimensões 3x3 com apenas 1 iteração.
3. Realizar um fechamento com EE em forma de quadrado com dimensões 3x3 com 7 iterações.
4. Recalcular o valor de moda para que seja realizado um *threshold*, porém sem considerar os pixels de valor 0.
5. Aplicar um *threshold* cujo limiar é dado por $\frac{\text{moda} \times 7}{65535}$.
6. Realizar uma abertura com um EE em forma de quadrado com dimensões 3x3 com 3 iterações.

Para finalizar o processo de segmentação, o pós-processamento descrito na seção 4.1.1 é realizado.

Figura 11 – *Dataset* Fluo-C2DL-MS

(a) Imagem “t000”.

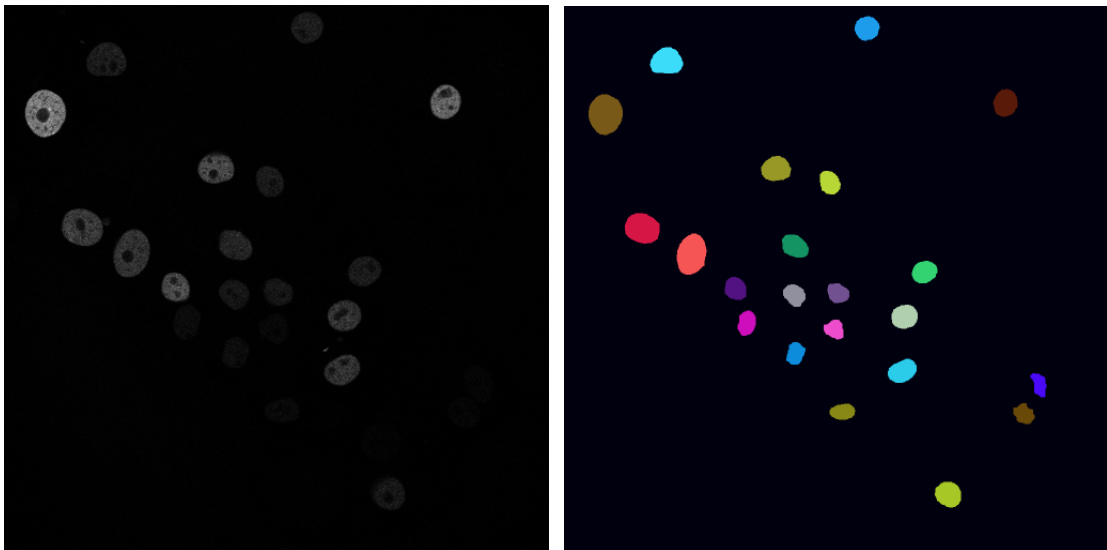
(b) Resultado da segmentação para a imagem “t000”.

4.1.7 *Dataset* Fluo-N2DH-GOWT1

Para realizar a segmentação deste *dataset*, apenas 2 operações foram necessárias:

1. Uma erosão com um EE com dimensões 3x3 em forma de quadrado com 2 iterações.
2. Um fechamento com um EE com dimensões 3x3 em forma de quadrado com 5 iterações.

Em seguida foi realizado o pós-processamento descrito na seção 4.1.1 para finalizar o processo de segmentação.

Figura 12 – *Dataset* Fluo-N2DH-GOWT1

(a) Imagem “t000”.

(b) Resultado da segmentação para a imagem “t000”.

4.1.8 Dataset PhC-C2DH-U373

Para realizar a segmentação deste *dataset*, primeiro é realizado um *top-hat* com um EE com dimensões 3x3 em forma de quadrado com 25 iterações. Em seguida é realizado um *threshold* cujo valor do limiar utilizado é gerado pela fórmula 4.4.

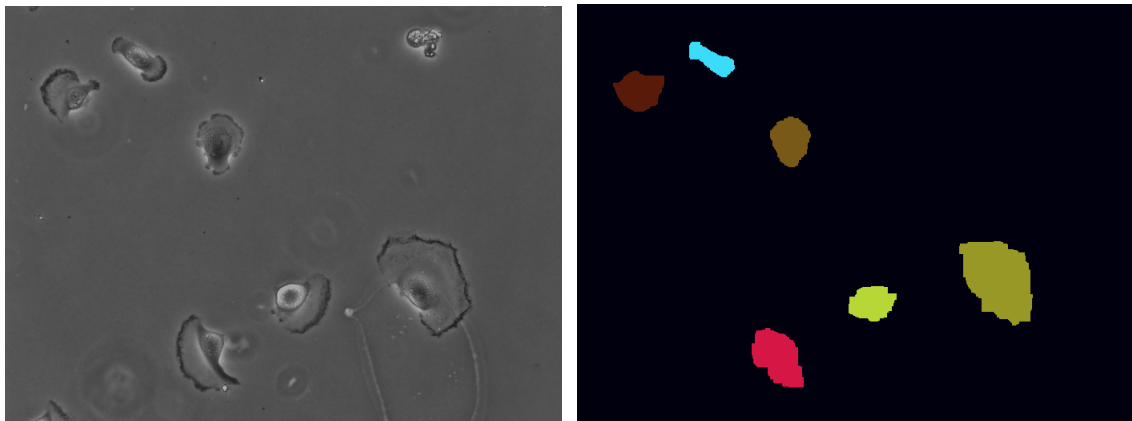
$$limiar = \frac{moda \times 5}{65535} \quad (4.4)$$

Em seguida são realizadas as seguintes operações:

1. Um fechamento com um EE com dimensões 3x3 em forma de quadrado com 5 iterações.
2. Uma erosão com um EE com dimensões 3x3 em forma de quadrado com 10 iterações.

Finalizando o processo de segmentação, o pós-processamento descrito na seção 4.1.1 foi realizado.

Figura 13 – Dataset PhC-C2DH-U373



(a) Imagem “t000”.

(b) Resultado da segmentação para a imagem “t000”.

4.2 Segunda Fase: Rotulação

O método de rotulação utilizado foi o de *searching and label propagation*. Este método foi escolhido pois, comparado ao método de *label equivalence* utilizado por Sousa (2015), demora mais para atingir o limite de rótulos disponíveis, o que o torna mais indicado para trabalhar com *datasets* com uma grande quantidade de componentes conexas. Abaixo está descrito como é realizado o processo de rotulação:

1. Inicializar o valor do rótulo atual.
2. Encontrar um pixel do tipo *foreground* não-rotulado na imagem, atribuir o valor de rótulo atual a ele e inseri-lo na pilha.

3. Enquanto a pilha não estiver vazia, retirar o pixel da pilha e verificar se seus pixels adjacentes do tipo *foreground* já foram rotulados. Se ainda não foram rotulados, atribuir o valor de rótulo atual e inseri-los na pilha.
4. Incrementar o valor do rótulo atual.
5. Enquanto houver pixel do tipo *foreground* não-rotulado na imagem, voltar para o passo 2.

Após rotular todas células, foram realizadas duas dilatações. Segundo [Sousa \(2015\)](#), estas dilatações são necessárias para que as células retornem a um tamanho próximo do original, pois, para que se pudesse ter o maior número de células possíveis sem estarem coladas umas às outras, os métodos de segmentação utilizados acabaram deixando as células com tamanhos muito menores que os originais. Isso previne a identificação errônea de duas células que se toquem sejam consideradas como uma única célula desde o início.

Após o processo de rotulação, é realizado o preenchimento da estrutura de dados CENTROID referente a cada imagem dos vídeos. Elas são necessárias para a fase de rastreamento, pois contém as informações necessárias de cada célula para fazer a vinculação durante a fase de rastreamento, tais como:

- a) O valor do rótulo da célula.
- b) O valor de *new_label*, que será utilizado para identificar a célula ao final do rastreamento.
- c) As coordenadas (x, y, z) do centroide da célula.
- d) Os valores do grau de entrada e saída.
- e) Os nós dos possíveis pais e filhos de uma célula.
- f) O índice do pai, que será utilizado para armazenar o valor do rótulo do pai das células resultantes de uma divisão celular.

4.3 Terceira Fase: Rastreamento

O rastreamento foi feito utilizando a abordagem de *tracking by detection* que, como visto anteriormente, consiste em realizar os passos de segmentação e vinculação de forma separada.

O rastreamento é baseado no conceito de grafos, no qual os vértices são representados pelas células, e as arestas representadas pela vinculação entre as células de uma imagem com as suas correspondentes na imagem consecutiva. Dessa forma, cada célula possui um grau de entrada, que representa a quantidade de possíveis pais, e um grau de saída, que representa a quantidade de possíveis filhos.

O processo de vinculação, segundo [Sousa \(2015\)](#), se deu através do cálculo da distância euclidiana e pela análise dos graus de entrada e saída de cada célula. Dessa forma estabeleceu-se as seguintes condições:

1. Se o grau de entrada da célula na imagem t fosse igual a 1, então ela receberia o mesmo valor de rótulo da célula correspondente a ela na imagem $t - 1$.
2. Se o grau de entrada da célula na imagem t fosse igual a 2, então seria necessário verificar os valores de rótulo de cada célula correspondente a ela na imagem $t - 1$.
 - a) Se os valores de rótulo de ambas as células na imagem $t - 1$ fossem iguais, então a célula na imagem t receberia o valor de rótulo de qualquer uma das células, uma vez que seria o mesmo valor para ambas.
 - b) Se os valores de rótulos de ambas as células na imagem $t - 1$ fossem diferentes, então seria necessário analisar os graus de saída da célula na imagem t .
 - i. Se o grau de saída da célula na imagem t for igual a 1, então ela recebe o valor de rótulo da célula correspondente na imagem $t - 1$ que tiver a menor distância.
 - ii. Se o grau de saída da célula na imagem t for igual a 0, então a célula deve receber os rótulos de ambas as células correspondentes na imagem $t - 1$. Nessa situação, após analisar as imagens, nós assumimos que as células, que antes estavam separadas em $t - 1$, se chocaram e que esse foi o último instante de aparição delas, por esse motivo a célula em t recebe os valores de rótulos de ambas as células.
 - iii. Se o grau de saída da célula na imagem t for igual a 2, então a célula em t deve receber os rótulos de ambas as células correspondentes na imagem $t - 1$, além disso, as células correspondentes a estas em $t + 1$ e nas posteriores devem receber os respectivos rótulos das células correspondentes em t e em $t - 1$. Assim como no caso do grau de saída igual a 0, nós assumimos que as células, que antes eram separadas em $t - 1$, se chocaram em t , porém voltaram a se separar novamente em $t + 1$, por esse motivo a célula em t recebe os valores dos rótulos de ambas as células, e as células em $t + 1$ recebem os valores respectivos de cada uma das células correspondentes a elas.

Além disso, para os casos de células que desapareciam momentaneamente nas imagens devido à segmentação, era necessário que estas fossem redesenhadas nas imagens. Mais detalhes de como essas reconstruções foram feitas, assim como as associações descritas anteriormente, podem ser vistos em [Sousa \(2015\)](#).

Uma das modificações realizadas no algoritmo de rastreamento com relação a [Sousa \(2015\)](#) foi no cálculo da distância euclidiana para poder estabelecer as relações de pai e filho

com base nos graus de entrada e saída. Antes de calcular a distância euclidiana, foi verificado se na imagem seguinte $t + 1$, nas coordenadas referentes ao centroide de cada célula na imagem t , havia alguma célula que pudesse ser a sua correspondente. Se houvesse, então a relação de pai e filho era estabelecida sem a necessidade de calcular a distância euclidiana. Isso foi feito para evitar fazer a vinculação de forma errônea em casos nos quais ocorrem divisão celular e os centroides das células filhas acabam ficando muito distantes do centroide da célula pai. Isso também previne que a célula seja associada a uma nova célula que esteja entrando na imagem e cujo centroide acaba ficando mais próximo do seu.

Tendo feito todas as associações, chega-se ao momento de identificar as ocorrências da divisão celular. Para auxiliar nessa identificação foi criada uma estrutura de dados chamada MITOSE que é responsável por armazenar as possíveis células resultantes de uma divisão celular com suas respectivas informações.

O preenchimento da estrutura de dados MITOSE foi feito analisando cada imagem do vídeo a fim de encontrar mais de uma célula que possuíssem o mesmo valor de *new_label*. Ao encontrar essas células, as seguintes informações de cada uma delas são inseridas na estrutura de dados MITOSE:

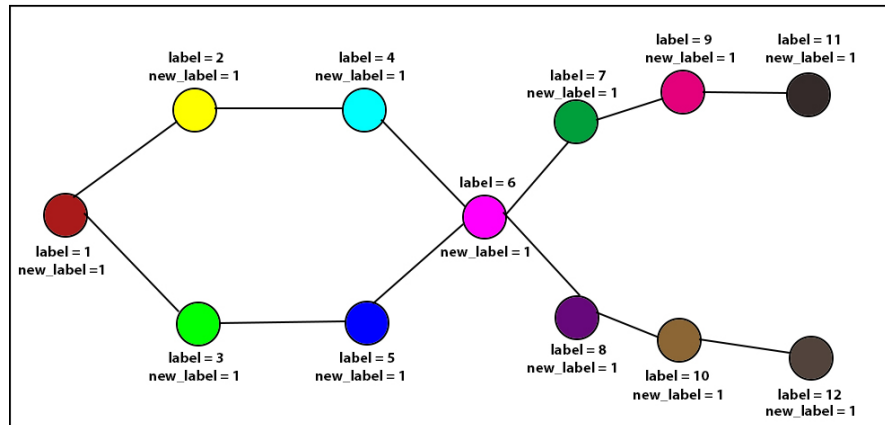
- a) O valor do rótulo atual.
- b) O valor de *new_label*.
- c) As coordenadas (x, y, z) do centroide da célula.
- d) O instante de tempo em que elas apareceram.

Tendo preenchido a estrutura de dados MITOSE, a resolução da divisão celular (ver Figura 14) é feita da seguinte maneira :

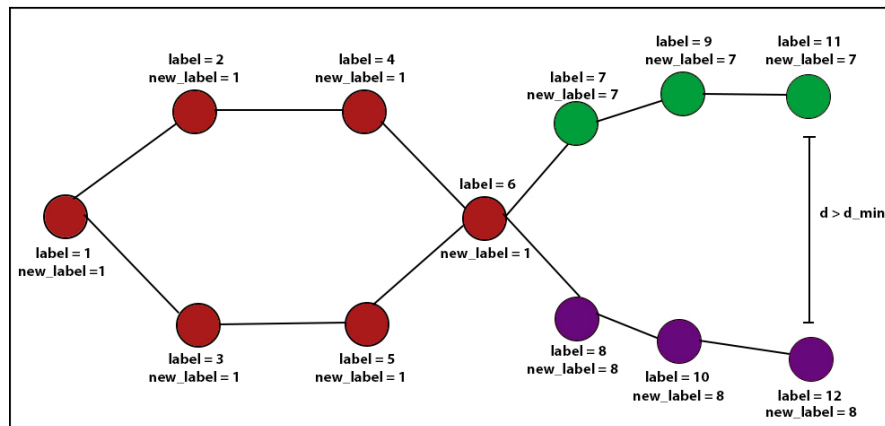
1. Identificar pares de células que possuam o mesmo valor de *new_label*.
2. Verificar por quantos instantes essas células se repetem sem que haja intervalo.
3. Caso o número de repetições seja maior que 2, verificar a distância (d) entre as células envolvidas no último instante de ocorrência. Escolhemos o valor mínimo de repetições como 2 devido ao fato da nossa abordagem seguir o princípio de que as células, ao se dividirem, começam a se afastar uma da outra, e adotamos o valor mínimo de 2 instantes de tempo para verificar o afastamento entre elas.
4. Se a distância (d) for maior que um valor mínimo determinado (d_{min}), é considerado divisão celular.

5. No caso de uma divisão celular, atribuir um novo valor de *new_label* para cada célula no primeiro instante de ocorrência; propagar esses novos valores para as células correspondentes nos instantes seguintes até o instante de última ocorrência; indicar o antigo valor de *new_label* como o pai das novas células.

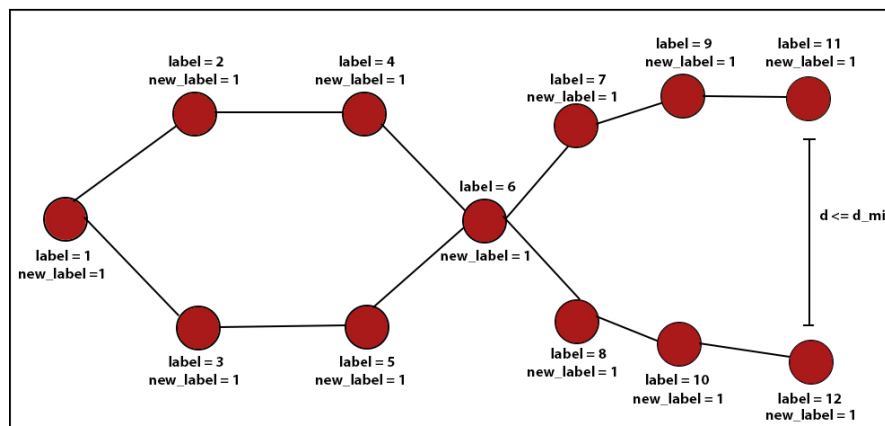
Figura 14 – Detecção de Divisão Celular



(a) Situação de possível divisão celular.



(b) Caso em que realmente é divisão celular.



(c) Caso em que não é divisão celular.

Ao finalizar o rastreamento, é criado um arquivo-texto contendo: o rótulo de cada célula; o tempo em que cada uma delas aparece pela primeira e pela última vez no vídeo; e o rótulo que identifica o pai dela, caso esta seja resultante de uma divisão celular, caso contrário é escrito o número zero para representar que não há pai, consequentemente, não havendo divisão.

4.4 Resultados Obtidos e Discussão

Os valores de acurácia para cada *dataset* podem ser vistos na Tabela 3. Os valores da fase de segmentação não foram tão bons, com valores variando de 9,7% a 85,1%, sendo o menor valor para a sequência 01 do *dataset* Fluo-C2DL-MSD, e o maior valor para a sequência 01 do *dataset* Fluo-N3DH-CHO. Além disso, a média da acurácia da segmentação ficou variando entre 27,9% e 81,6%, sendo que o *dataset* Fluo-C3DL-MDA231 teve uma média de 61,3%, o *dataset* Fluo-C3DH-H157 de 68%, o *dataset* Fluo-N3DH-SIM de 63,7%, o *dataset* Fluo-N3DH-CHO de 81,6%, o *dataset* Fluo-C2DL-MSD de 27,9%, o *dataset* Fluo-N2DH-GOWT1 de 64,3%, e o *dataset* PhC-C2DH-U373 de 38,8%.

Embora os valores de acurácia da fase de segmentação, de uma forma geral, não tenham sido tão bons, o algoritmo proposto para a fase de rastreamento chegou a obter valores de acurácia acima de 90% para os *datasets* 3D. O menor valor obtido foi na sequência 06 do *dataset* Fluo-N3DH-SIM com um valor de acurácia de 87,2%, e o maior valor obtido foi de 100% na sequência 01 deste mesmo *dataset*. Ambos os *datasets* 3D tiveram uma média de acurácia em torno dos 90% na fase de rastreamento (ver Tabela 3), sendo que o *dataset* Fluo-C3DL-MDA231 teve uma média de 90,1%, o *dataset* Fluo-C3DH-H157 de 94%, o *dataset* Fluo-N3DH-SIM de 93,3%, e o *dataset* Fluo-N3DH-CHO de 94,3%. Porém, o mesmo não pode ser dito para os *datasets* 2D cujas médias de acurácia para o rastreamento foram de 58,8% para o *dataset* Fluo-C2DL-MSD, de 85,9% para o *dataset* Fluo-N2DH-GOWT1, e de 70,2% para o *dataset* PhC-C2DH-U373.

Dessa forma, é possível perceber que o algoritmo de rastreamento proposto possui um bom funcionamento para os *datasets* 3D enquanto que para os *datasets* 2D ele não é tão efetivo, embora ainda consiga realizar o rastreamento e gerar resultados. Isso pode estar relacionado à quantidade de células identificadas, pois, embora os métodos escolhidos para identificação da célula, ou seja, para a segmentação da célula, tenham atingido o objetivo inicial, que era ter o maior número de células sem estarem grudadas umas nas outras no instante 0 do vídeo, os métodos podem não ter sido os melhores para os vídeos 2D como um todo, como pode ser visto nos resultados obtidos para a segmentação (ver Tabela 3). Porém, para fazer a verificação das células identificadas seria necessário analisar cada uma das imagens resultantes da fase de segmentação com a imagem correspondente do *ground truth* do rastreamento, visto que é disponibilizado o *ground truth* para todas as imagens do vídeo, diferente do *ground truth* da segmentação, que disponibiliza apenas parte das imagens. Dessa forma, seria possível verificar

Tabela 3 – Resultados Obtidos pelo Algoritmo Proposto

Dataset	Sequência	SEG	TRA
Fluo-C3DL-MDA231	01	0,574	0,884
	02	0,653	0,918
	Média	0,613	0,901
Fluo-C3DH-H157	01	0,790	0,990
	02	0,570	0,891
	Média	0,680	0,940
Fluo-N3DH-SIM	01	0,710	1,000
	02	0,692	0,933
	03	0,617	0,960
	04	0,611	0,947
	05	0,647	0,885
	06	0,547	0,872
	Média	0,637	0,933
Fluo-N3DH-CHO	01	0,851	0,971
	02	0,781	0,916
	Média	0,816	0,943
Fluo-C2DL-MSC	01	0,097	0,490
	02	0,461	0,687
	Média	0,279	0,588
Fluo-N2DH-GOWT1	01	0,513	0,872
	02	0,773	0,846
	Média	0,643	0,859
PhC-C2DH-U373	01	0,352	0,759
	02	0,424	0,645
	Média	0,388	0,702

de fato quantas células foram identificadas corretamente no vídeo por completo.

Comparando-se os resultados obtidos para segmentação com os obtidos para o rastreamento, podemos observar que mesmo tendo resultados de segmentação abaixo de 100% é possível ter um resultado de rastreamento com 100% de acerto. Um exemplo disso é na sequência 01 do *dataset* Fluo-N3DH-SIM que obteve um resultado de 71% para a fase de segmentação e de 100% para o rastreamento. Isso se deve ao fato da segmentação levar em consideração o tamanho da célula, enquanto que a fase de rastreamento leva em consideração as células identificadas e rastreadas. No caso deste *dataset*, todas as células foram identificadas e rastreadas corretamente, por isso o valor de 100% para o rastreamento, porém, os tamanhos destas células diferem dos seus tamanhos originais, ocasionando o valor de 71% para a segmentação, pois para se obter um valor de 100% para a segmentação, as células identificadas teriam que ter exatamente o mesmo tamanho das células originais.

Foi feita também uma comparação entre os resultados obtidos pelo algoritmo proposto e os resultados dos competidores do *Cell Tracking Challenge* (ver Tabela 4). A comparação foi feita para todos os *datasets* utilizados neste trabalho, exceto para o *dataset* Fluo-N3DH-SIM,

pois os resultados para este *dataset* não estão disponíveis para comparação. Lembrando que o *dataset* Fluo-N3DH-SIM, que não está disponível, é diferente do *dataset* Fluo-N3DH-SIM+, que está disponível.

Os resultados dos competidores para comparação foram tirados de [Ulman et al. \(2017\)](#), que disponibiliza os resultados para cada uma das medidas utilizadas: medida da fase de segmentação (SEG); medida da fase de rastreamento (TRA); e uma medida de desempenho geral (OP), que foi utilizada para permitir uma comparação direta dos métodos dos competidores como um todo e é dada pela Fórmula 4.5.

$$OP = 0,5 \cdot (SEG + TRA) \quad (4.5)$$

Porém, apenas os resultados dos três melhores competidores foram disponibilizados por [Ulman et al. \(2017\)](#) e, não necessariamente, os três primeiros colocados para uma medida são os mesmos três primeiros colocados das outras medidas. Por esse motivo, nem todos os competidores possuem valores para todas as medidas, como pode ser visto na Tabela 4.

Comparado aos competidores do *Cell Tracking Challenge*, o algoritmo implementado obteve bons resultados de acurácia para a fase de rastreamento para os *datasets* 3D (ver Tabela 4). O resultado de TRA obtido para o *dataset* Fluo-C3DL-MDA231 foi melhor que de todos os outros competidores. O resultado obtido para o *dataset* Fluo-N3DH-CHO está entre os três melhores. E, embora não tenha ficado entre os três primeiros colocados, assim como os *datasets* anteriores, o *dataset* Fluo-C3DH-H157 obteve uma média de resultado próximo dos três primeiros colocados, ficando atrás do terceiro colocado por menos de 2% de diferença. Já em relação aos *datasets* 2D, nenhum deles obtiveram resultados próximos ou melhores que os dos competidores do *Cell Tracking Challenge*.

Fazendo um comparativo em relação à fase de segmentação, percebe-se que nenhum dos *datasets*, tanto 2D quanto 3D, obtiveram resultados que fossem próximos ou melhores que os dos competidores, excetuando-se o *dataset* Fluo-C3DL-MDA231 que obteve uma média de 61,3%, ficando dentre os três primeiros colocados.

Ainda em relação ao *dataset* Fluo-C3DL-MDA231, é possível fazer uma comparação dos resultados obtidos neste trabalho com os obtidos por [Sousa \(2015\)](#). Comparando os resultados da segmentação, tem-se que [Sousa \(2015\)](#) obteve uma acurácia de 0,579 para o vídeo 01, e de 0,653 para o vídeo 02, resultando em uma média de 0,616 para a fase de segmentação. Já neste trabalho obteve-se uma acurácia de 0,574 para o vídeo 01, e de 0,653 para o vídeo 02, resultando em uma média de 0,613.

Fazendo uma comparação também em relação à fase de rastreamento, tem-se que [Sousa \(2015\)](#) obteve uma acurácia de 0,882 para o vídeo 01, e de 0,914 para o vídeo 02, resultando em uma média de 0,898. Já neste trabalho obteve-se uma acurácia de 0,884 para o vídeo 01, e de 0,918 para o vídeo 02, resultando em uma média de 0,901.

Embora a média de segmentação obtida neste trabalho tenha sofrido uma queda de 0,003 em relação à obtida por [Sousa \(2015\)](#), a média de rastreamento obteve um aumento de 0,003.

Tabela 4 – Comparação Entre os Resultados do Algoritmo Proposto e dos Competidores do *Cell Tracking Challenge*

Dataset	Group	SEG	TRA	OP
Fluo-C3DL-MDA231	KTH-SE	0,631	0,883	0,757
	LEID-NL	0,625	0,865	0,745
	IMCB-SG	0,504	-	0,659
	HD-Har-GE	-	0,830	-
	Algoritmo Proposto	0,613	0,901	0,757
Fluo-C3DH-H157	KTH-SE	0,888	0,987	0,938
	LEID-NL	0,773	-	-
	CUNI-CZ	-	0,976	0,870
	HD-Har-GE	0,816	0,954	0,885
	Algoritmo Proposto	0,680	0,940	0,810
Fluo-N3DH-CHO	MU-CZ	0,917	-	0,912
	KTH-SE	0,899	0,953	0,926
	HD-Har-GE	0,898	0,914	0,906
	CUNI-CZ	-	0,935	-
	Algoritmo Proposto	0,816	0,943	0,879
Fluo-C2DL-MSC	KTH-SE	0,590	0,763	0,676
	FR-Ro-GE	0,582	0,691	0,636
	UZH-CH	0,465	-	-
	NOTT-UK	-	0,645	0,546
	Algoritmo Proposto	0,279	0,588	0,433
Fluo-N2DH-GOWT1	KTH-SE	0,927	0,976	0,951
	LEID-NL	0,893	-	0,902
	CUNI-CZ	0,887	0,916	0,902
	KIT-GE	-	0,925	-
	Algoritmo Proposto	0,643	0,859	0,751
PhC-C2DH-U373	FR-Ro-GE	0,920	0,981	0,951
	FR-Be-GE	0,826	0,965	0,896
	KTH-SE	0,795	0,978	0,886
	Algoritmo Proposto	0,388	0,702	0,545

4.4.1 Dificuldades

Além dos 7 *datasets* apresentados neste trabalho, mais 4 *datasets* (Fluo-N2DL-HeLa, Fluo-N3DH-SIM+, Fluo-N3DH-CE e DIC-C2DH-HeLa) foram testados. Porém, não foi possível obter resultados consistentes para estes *datasets* com o algoritmo apresentado.

Algumas das células do *dataset* Fluo-N2DL-HeLa se encaixam no problema de célula que desaparece momentaneamente por acabar desaparecendo junto com o *background* da imagem durante o processo de segmentação. Porém, estas células acabam não sendo reconstruídas durante o processo de rastreamento por não satisfazerem as condições necessárias impostas pelo algoritmo para a realização da reconstrução. As condições para reconstrução são:

- a) O grau de saída da célula pai da célula a ser reconstruída ser igual a 0.
- b) O grau de entrada da célula filha da célula a ser reconstruída ser igual a 0.

Uma grande quantidade de células do *dataset* DIC-C2DH-HeLa encontram-se coladas umas nas outras e misturadas ao *background* da imagem, o que acaba dificultando o processo de segmentação. Se tentar manter as células que se misturam com o *background* da imagem, não será possível segmentar as células que estão coladas. E se tentar separar as células que estão coladas, as células que se misturam com o *background* da imagem serão eliminadas junto com o *background* durante a fase de segmentação. Ambas as situações acabam gerando inconsistências no processo de rastreamento.

No *dataset* Fluo-N3DH-SIM+ temos situações em que uma mesma célula acaba se dividindo mais de uma vez com o passar do tempo. O algoritmo implementado neste trabalho não foi projetado para trabalhar com estes tipos de situações, e, por esse motivo, acaba gerando inconsistências. A inconsistência gerada é a de ter e, ao mesmo tempo, não ter ocorrido divisão celular, pois nem todas as células filhas recebem novos rótulos para serem identificadas, continuando com o mesmo rótulo da célula pai.

O *dataset* Fluo-N3DH-CE recai na mesma situação do *dataset* Fluo-N3DH-SIM+, em que uma mesma célula acaba se dividindo mais de uma vez com o passar do tempo. Além disso, à medida que as células vão se dividindo, seus tamanhos vão se alterando, o que dificulta em adotar um padrão de identificação das células para diferenciá-las de ruídos que possuam um tamanho consideravelmente alto. Outro fator a se levar em consideração é que este *dataset* possui uma grande quantidade de células coladas umas nas outras, o que dificulta no processo de segmentação.

5

Conclusão

Com a alteração do método de rotulação, foi possível executar o algoritmo de rastreamento em 7 dos 11 *datasets* utilizados, diferente de Sousa (2015), que tinha limitações em *datasets* com muitas células por não conseguir rotulá-las antes do término da quantidade de rótulos disponíveis.

Embora o valor médio do método de segmentação utilizado para o *dataset* Fluo-C3DL-MDA231 tenha tido uma queda de 0,003 em relação ao de Sousa (2015), o método de rastreamento junto com a detecção da divisão celular e a entrada e saída de novas células gerou um aumento de 0,003 no valor médio do rastreamento.

Embora os valores de acurácia da fase de segmentação para todos os *datasets*, tanto 2D quanto 3D, não tenham sido satisfatórios, foi possível obter resultados satisfatórios para alguns dos *datasets* na fase de rastreamento. Os valores médios de SEG ficaram entre 61,3% e 81,6% para os *datasets* 3D, e entre 27,9% e 64,3% para os 2D. Enquanto que os valores médios de TRA ficaram entre 90,1% e 94,3% para os *datasets* 3D, e entre 58,8% e 85,9% para os 2D. Isso pode ser facilmente explicado pelo fato da medida SEG levar em consideração o índice de Jaccard que é calculado com base no tamanho total de cada célula, enquanto que a medida de TRA não leva em consideração o tamanho das células, mas sim a quantidade de operações necessárias para transformar o grafo gerado pelo algoritmo implementado no grafo do *ground truth*.

Dentre todos os algoritmos de segmentação utilizados, o único que obteve um resultado satisfatório foi o do *dataset* Fluo-C3DL-MDA231. Ele obteve um resultado médio de 61,3%, ficando entre os 3 primeiros colocados quando comparado aos competidores do *Cell Tracking Challenge* cujos primeiro, segundo e terceiro colocados obtiveram resultados médios de 63,1%, 62,5% e de 50,4%, respectivamente.

Com relação à fase de rastreamento, nenhum dos *datasets* 2D obtiveram resultados satisfatórios. Todos eles foram aquém do esperado, com valores médios inferiores aos dos 3 primeiros colocados do *Cell Tracking Challenge*, com uma diferença para o terceiro colocado de:

5,7% para o *dataset* Fluo-C2DL-MSD, 5,7% para o *dataset* Fluo-N2DH-GOWT1, e 26,3% para o *dataset* PhC-C2DH-U373.

Em compensação, os valores para a fase de rastreamento dos *datasets* 3D foram bastante significativos. O *dataset* Fluo-C3DL-MDA231 superou as expectativas ficando na frente de todos os competidores, com um valor de TRA médio de 90,1%. O *dataset* Fluo-N3DH-CHO com um valor de TRA médio de 94,3% ficou em segundo colocado com uma diferença de apenas 1% do primeiro colocado. O *dataset* Fluo-C3DH-H157 embora não tenha ficado entre os 3 primeiros colocados, a diferença para o terceiro colocado foi de 1,4%, sendo seu valor de TRA médio de 94% enquanto que o do terceiro colocado de 95,4%.

Embora não tenha sido possível comparar os valores obtidos para o *dataset* Fluo-N3DH-SIM com o dos competidores, este *dataset* também obteve um valor de TRA satisfatório, com um valor médio de acurácia de 93,3%, e em uma das sequências (sequência 01) obteve resultado de 100%.

Mesmo tendo obtido alguns valores satisfatórios, o algoritmo ainda precisa de melhorias, visto que ainda há *datasets* nos quais o algoritmo não executa como o esperado. Embora alguns problemas, como o de desaparecimento momentâneo de células e o de células coladas umas às outras, já possuam uma solução, esta não funciona para todos os *datasets*. Além disso, em alguns *datasets*, as células se dividem mais de uma vez com o passar do tempo, e o algoritmo implementado não foi desenvolvido visando estes tipos de ocorrência.

Por esse motivo, como trabalhos futuros, pretende-se:

- a) Alterar a abordagem utilizada para resolver o problema de desaparecimento momentâneo de células, para que todas as ocorrências dessa situação possam ser resolvidas.
- b) Criar novos métodos de segmentação que melhor se adaptem às situações em que muitas células estejam coladas umas nas outras e que se misturam ao *background* da imagem.
- c) Alterar a abordagem de rastreamento para atender aos casos em que uma mesma célula acaba se dividindo mais de uma vez com o passar do tempo.
- d) Criar um método para verificar a quantidade de células que foram identificadas corretamente durante a fase de segmentação.
- e) Implementar o modelo matemático descrito neste trabalho e utilizá-lo para auxiliar no rastreamento. Com a implementação do modelo, pretende-se melhorar os resultados do rastreamento visto que será possível estimar a distância máxima que cada tipo de célula pode se mover em relação ao tempo anterior, diferente da abordagem atual em que foi fixado um único valor máximo de distância para todos os tipos de células.

Referências

- BARRETT, H. H. Fundamentals of the radon transform. In: VIERGEVER, M. A.; TODD-POKROPEK, A. (Ed.). *Mathematics and Computer Science in Medical Imaging*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1988. p. 105–125. ISBN 978-3-642-83306-9. Citado na página 36.
- CADART, C. et al. Exploring the function of cell shape and size during mitosis. *Developmental Cell*, v. 29, n. 2, p. 159 – 169, 2014. ISSN 1534-5807. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1534580714002299>. Citado na página 15.
- DANTAS, D. O.; LEAL, H. D. P.; SOUSA, D. O. B. FAST MULTIDIMENSIONAL IMAGE PROCESSING WITH OPENCL. *IEEE International Conference on Image Processing (ICIP)*, p. 1779–1783, 2016. Citado na página 46.
- DASS, R.; PRIYANKA; DEVI, S. Image segmentation techniques. *International Journal of Electronics & Communication Technology*, v. 3, n. 1, p. 66–70, 2012. Citado na página 21.
- DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern Classification (2nd Edition)*. 2. ed. [S.l.]: Wiley-Interscience, 2000. Hardcover. ISBN 0471056693. Citado na página 35.
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. 2. ed. [S.l.]: Prentice Hall, 2002. ISBN 0-201-18075-8. Citado na página 37.
- HE, L.; CHAO, Y.; SUZUKI, K. An efficient first-scan method for label-equivalence-based labeling algorithms. *Pattern Recognition Letters*, Elsevier B.V., v. 31, n. 1, p. 28–35, 2010. ISSN 01678655. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167865509002244>. Citado na página 24.
- INCA. *Estimativa 2018: incidência de câncer no Brasil*. [S.l.: s.n.], 2017. ISBN 978-85-7318-362-7. Citado na página 16.
- KHAN, A. M.; RAVI, S. Image Segmentation Methods: A Comparative Study. *International Journal of Soft Computing and Engineering (IJSCE)*, v. 3, n. 4, p. 84–92, 2013. ISSN 10179909. Citado na página 21.
- KUHN, H. W. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, v. 2, p. 83–97, 1955. Citado na página 38.
- LEE, J. Insights into cell motility provided by the iterative use of mathematical modeling and experimentation. *AIMS Biophysics*, v. 5, 2018. ISSN 2377-9098. Citado 2 vezes nas páginas 18 e 19.
- LEE, J. et al. Principles of locomotion for simple-shaped cells. *Nature*, v. 362, n. 6416, p. 167—171, March 1993. ISSN 0028-0836. Disponível em: <https://doi.org/10.1038/362167a0>. Citado 2 vezes nas páginas 19 e 20.
- LENZ, P. *Cell Motility*. [S.l.]: Springer, 2008. ISBN 978-0-387-73049-3. Citado 2 vezes nas páginas 13 e 14.

- MAGNUSSON, K. E. G.; JALDEN, J. *Tracking of non-brownian particles using the Viterbi algorithm*. 2015. Citado na página 35.
- MAGNUSSON, K. E. G. et al. Global linking of cell tracks using the viterbi algorithm. *IEEE Transactions on Medical Imaging*, v. 34, n. 4, p. 911 – 929, 2015. Citado na página 35.
- MAŠKA, M. et al. A benchmark for comparison of cell tracking algorithms. *Bioinformatics*, v. 30, n. 11, p. 1609–1617, 2014. ISSN 1367-4803. Disponível em: <<http://bioinformatics.oxfordjournals.org/cgi/doi/10.1093/bioinformatics/btu080>>. Citado 2 vezes nas páginas 26 e 33.
- MASSOUDI, A.; SEMENOVICH, D.; SOWMYA, A. Cell tracking and mitosis detection using splitting flow networks in phase-contrast imaging. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, p. 5310–5313, 2012. ISSN 1557170X. Citado na página 15.
- MEIJERING, E.; DZYUBACHYK, O.; SMAL, I. Chapter nine - methods for cell and particle tracking. In: CONN, P. M. (Ed.). *Imaging and Spectroscopic Analysis of Living Cells*. Academic Press, 2012, (Methods in Enzymology, v. 504). p. 183 – 200. Disponível em: <<http://www.sciencedirect.com/science/article/pii/B9780123918574000094>>. Citado na página 26.
- MISHCHENKO, Y. A fast algorithm for computation of discrete Euclidean distance transform in three or more dimensions on vector processing. 2012. Citado na página 35.
- OTSU, N. A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, v. 9, n. 1, p. 62–66, 1979. Disponível em: <<http://dx.doi.org/10.1109/TSMC.1979.4310076>>. Citado na página 37.
- SAHOO, P.; WILKINS, C.; YEAGER, J. Threshold selection using renyi's entropy. *Pattern Recognition*, v. 30, n. 1, p. 71 – 84, 1997. ISSN 0031-3203. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0031320396000659>>. Citado na página 38.
- SCHNEIDER, C. A.; RASBAND, W. S.; ELICEIRI, K. W. Nih image to imagej: 25 years of image analysis. *Nat Meth*, Nature Publishing Group, a division of Macmillan Publishers Limited. All Rights Reserved., v. 9, n. 7, p. 671–675, jul. 2012. ISSN 15487091. Disponível em: <<http://dx.doi.org/10.1038/nmeth.2089>>. Citado na página 36.
- SERTEL, O. et al. Microscopic image analysis for quantitative characterization of muscle fiber type composition. *Computerized Medical Imaging and Graphics*, v. 35, n. 7, p. 616 – 628, 2011. ISSN 0895-6111. Whole Slide Image Process. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0895611111000206>>. Citado na página 34.
- SMOCHINA, C.; HERGHELEGIU, P.; MANTA, V. Image processing techniques used in microscopic image segmentation. *Bul. Inst. Polit. Iași, t. LVII (LXI), f.2*, n. Lxi, 2011. Citado na página 22.
- SOLORZANO, C. O. de. *Cell Tracking Challenge Description*. 2017. Disponível em: <<http://www.celltrackingchallenge.net/description.html>>. Acesso em: 30 nov 2017. Citado na página 16.
- SOLORZANO, C. O. de. *Cell Tracking Challenge Evaluation*. 2017. Disponível em: <<http://www.celltrackingchallenge.net/evaluation.html>>. Acesso em: 29 nov 2017. Citado na página 26.

SOLORZANO, C. O. de. *Cell Tracking Challenge Submission of Results*. 2017. Disponível em: <<http://www.celltrackingchallenge.net/submission-of-results.html>>. Acesso em: 29 nov 2017. Citado na página 26.

SOLORZANO, C. O. de. *Datasets*. 2017. Disponível em: <<http://www.celltrackingchallenge.net/datasets.html>>. Acesso em: 15 jun 2017. Citado na página 16.

SONKA, M.; HLAVAC, V.; BOYLE, R. *Image Processing, Analysis, and Machine Vision, International Student Edition*. [S.l.]: Thomson, 2008. ISBN 0-495-24438-4. Citado na página 25.

SOUSA, D. O. B. *Rastreamento de células em imagens de microscopia confocal*. 2015. Trabalho de Conclusão de Curso (Bacharel em Ciência da Computação), UFS (Universidade Federal de Sergipe), São Cristóvão, Brasil. Citado 14 vezes nas páginas 7, 8, 13, 17, 25, 27, 30, 54, 55, 56, 61, 62, 64 e 70.

ULMAN, V. et al. An objective comparison of cell tracking algorithms. *Nature Methods*, v. 14, p. 1141–1152, 2017. Citado 3 vezes nas páginas 33, 61 e 70.

VALLOTTON, P. et al. Tracking retrograde flow in keratocytes: news from the front. *Molecular biology of the cell*, v. 16, p. 1223–1231, 2005. Citado na página 19.

VO, B.-N.; MA, W.-K. The gaussian mixture probability hypothesis density filter. *Trans. Sig. Proc.*, IEEE Press, Piscataway, NJ, USA, v. 54, n. 11, p. 4091–4104, nov. 2006. ISSN 1053-587X. Disponível em: <<http://dx.doi.org/10.1109/TSP.2006.881190>>. Citado na página 35.

YILMAZ, A.; JAVED, O.; SHAH, M. Object tracking: A survey. *ACM Comput. Surv.*, v. 38, n. 4, p. 13, 2006. Citado na página 25.

ZIEBERT, F.; ARANSON, I. S. Computational approaches to substrate-based cell motility. *npj Computational Materials*, 2016. Citado na página 21.

ZIMMER, C. et al. On the digital trail of mobile cells. *IEEE Signal Processing Magazine*, v. 23, n. 3, p. 54–62, 2006. ISSN 10535888. Citado 3 vezes nas páginas 13, 14 e 15.

Anexos

ANEXO A – Descrição dos Datasets

Neste anexo são apresentados os *datasets* utilizados na execução deste trabalho, cujas descrições são dadas na Tabela 5 e uma representação visual de cada um deles também pode ser vista na Figura 16. Para informações mais detalhadas a respeito de cada *dataset*, consultar [Ulman et al. \(2017\)](#).

A nomenclatura dos *datasets* segue a convenção “Prefixo-LNDR-Sufixo”, onde:

- a) Prefixo: identifica a modalidade de microscopia. Podendo ser de fluorescência (Fluo), de contraste de fase (PhC) ou de contraste de interferência diferencial (DIC).
- b) LNDR: identifica o método de rotulação (L), que pode ser citoplasmático (C) ou nuclear (N); a dimensão (ND), que pode ser bidimensional (2D) ou tridimensional (3D); e a resolução (R), que pode ser alta (H) ou baixa (L).
- c) Sufixo: identifica a linhagem das células.

Cada dataset possui um valor de campo de interesse (FoI) que delimita a área da imagem a ser analisada. Segundo [Sousa \(2015\)](#), isso serve para que apenas objetos que tenham adentrado substancialmente na imagem sejam considerados durante a análise. O valor de FoI de cada *dataset* pode ser visto na Tabela 5 e uma representação das distâncias entre cada valor de FoI e a borda da imagem pode ser vista na Figura 15.

Figura 15 – Representação das distâncias do FoI

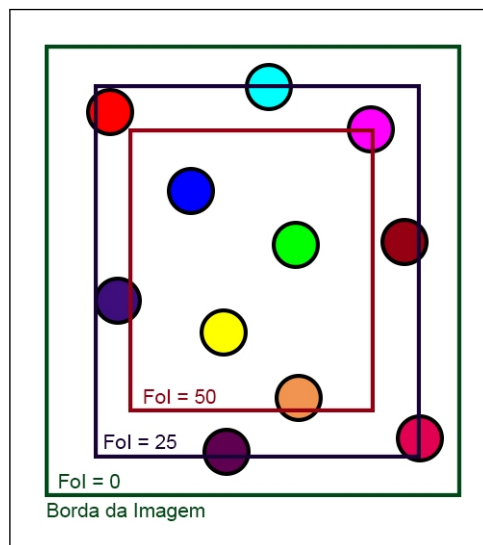
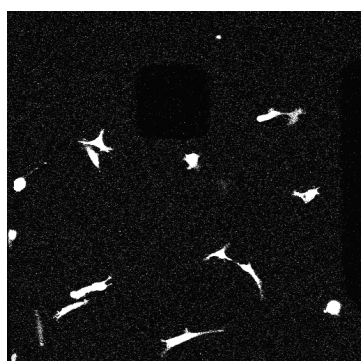
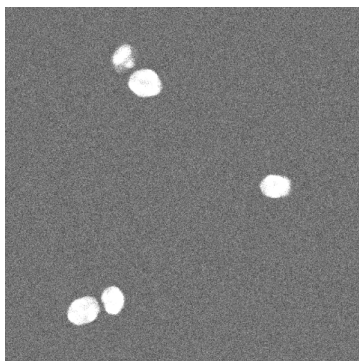


Tabela 5 – Características dos *Datasets*

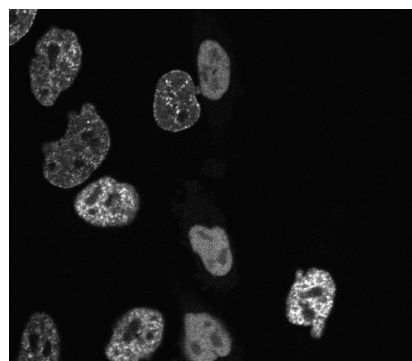
<i>Dataset</i>	Descrição	Vídeo	FoI (pixel)	Quantidade de Imagens	Dimensões das Imagens	Bits por pixel	Resolução dos Pixels (μm)	Intervalo de Tempo (min)
Fluo-C3DL-MDA231	Células de câncer de mama MDA231 infectadas por um vetor pMSCV incluindo uma sequência GFP imersas em uma matriz de colágeno.	1	25	12	512x512x30	16	1,24x1,24x6	80
		2	25	12	512x512x30	16	1,24x1,24x6	80
Fluo-C3DH-H157	Células de carcinoma de pulmão humano H157 transfectadas com GFP imersas em uma matriz de matrigel.	1	50	60	992x832x35	16	0,12x0,12x0,5	1
		2	50	60	992x832x80	16	0,12x0,12x0,5	2
Fluo-N3DH-SIM	Imagens simuladas com núcleos de células da linhagem HL60 rotuladas fluorescentemente migrando em uma matriz de colágeno.	1	0	51	640x640x70	16	0,125x0,125x0,2	28,8 ou 57,6
		2	0	51	640x560x70	16	0,125x0,125x0,2	28,8 ou 57,6
		3	0	51	640x640x70	16	0,125x0,125x0,2	28,8 ou 57,6
		4	0	50	569x640x70	16	0,125x0,125x0,2	28,8 ou 57,6
		5	0	97	500x588x70	16	0,125x0,125x0,2	28,8 ou 57,6
		6	0	100	480x270x70	16	0,125x0,125x0,2	28,8 ou 57,6
Fluo-N3DH-CHO	Núcleos de células de hamster chinês corados com cromatina por transfecção com PCNA-GFP.	1	50	92	512x443x5	8	0,2x0,2x1	9,5
		2	50	92	512x443x5	8	0,2x0,2x1	9,5
Fluo-C2DL-MSC	Células-tronco mesenquimais de rato em um substrato de poliacrilamida plano corado por transfecção estável com Actin-GFP.	1	50	48	992x832	16	0,3x0,3	20
		2	50	48	1200x782	16	0,398x0,398	30
Fluo-N2DH-GOWT1	Núcleos multipotentes de células-tronco embrionárias de camundongos corados pela cromatina por transfecção estável com histona H2B-GFP.	1	50	92	1024x1024	8	0,24x0,24	5
		2	50	92	1024x1024	8	0,24x0,24	5
PhC-C2DH-U373	Células de glioblastoma-astrocitoma U373 em um substrato 2D de poliacrilamida.	1	50	115	696x520	8	0,65x0,65	15
		2	50	115	696x520	8	0,65x0,65	15
Fluo-N2DL-HeLa	Células HeLa estavelmente transfectadas com H2B-GFP, obtida de um câncer cervical.	1	25	92	1100x700	16	0,645x0,645	30
		2	25	92	1100x700	16	0,645x0,645	30
Fluo-N3DH-SIM+	Simulação de células inspiradas em grande parte pelos núcleos HL60 e HeLa corados com corante Hoechst 33342.	1	0	150	639x349x59	16	0,125x0,125x0,200	29
		2	0	80	652x642x59	16	0,125x0,125x0,200	29
Fluo-N3DH-CE	Fase inicial de <i>C. elegans</i> em desenvolvimento de embriões com núcleos corados por transfecção de GFP.	1	50	250	708x512x35	8	0,09x0,09x1,0	1 ou 1,5
		2	50	250	712x512x31	8	0,09x0,09x1,0	1 ou 1,5
DIC-C2DH-HeLa	Células HeLa em um substrato de vidro plano.	1	50	84	512x512	8	0,19x0,19	10
		2	50	84	512x512	8	0,19x0,19	10

Figura 16 – Imagens Representativas dos *Datasets*

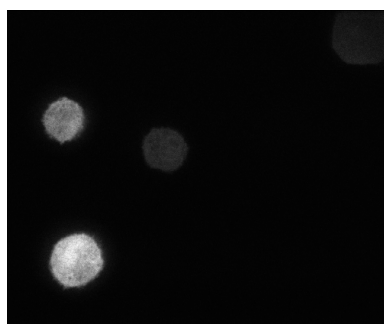
(a) Fluo-C3DL-MDA231



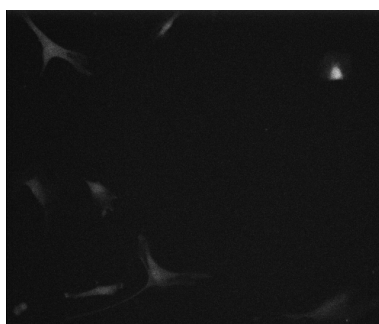
(b) Fluo-N3DH-SIM



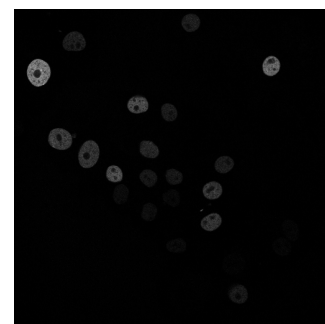
(c) Fluo-N3DH-CHO



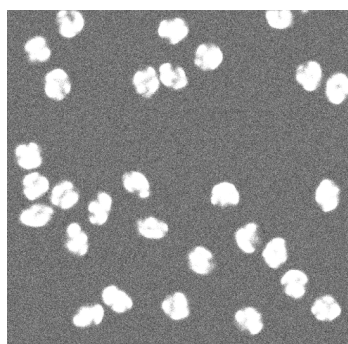
(d) Fluo-C3DH-H157



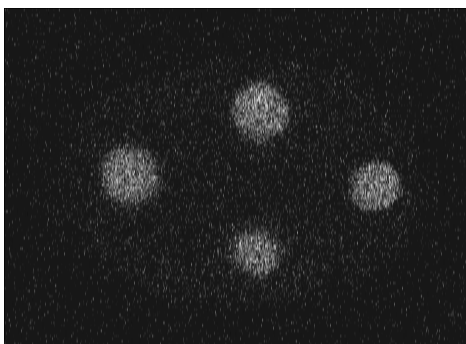
(e) Fluo-C2DL-MS



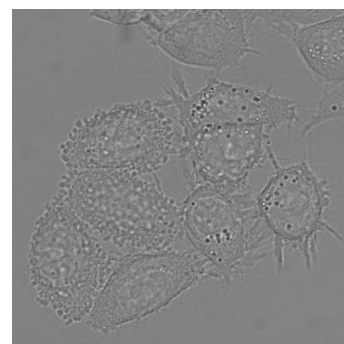
(f) Fluo-N2DH-GOWT1



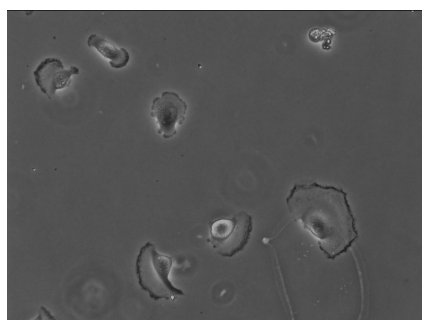
(g) Fluo-N3DH-SIM+



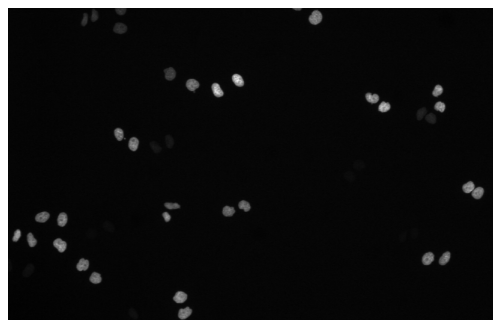
(h) Fluo-N3DH-CE



(i) DIC-C2DH-HeLa



(j) PhC-C2DH-U373



(k) Fluo-N2DL-HeLa